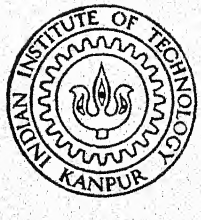
A NUMERICAL SOLUTION OF THE INVERSE PROBLEM FOR HYDRAULIC TURBOMACHINES USING A SINGULARITY METHOD

By
NARENDRA KUMAR JAIN

ME
1977
M
JAI
NUM

me/1977/m
J199n



DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR
NOVEMBER, 1977

A NUMERICAL SOLUTION OF THE INVERSE PROBLEM FOR HYDRAULIC TURBOMACHINES USING A SINGULARITY METHOD

A Thesis Submitted
In Partial Fulfilment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY

19[illegible]

By
NARENDRA KUMAR JAIN

to the
DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR
NOVEMBER, 1977

I.I.T. KANPUR
CENTRAL LIBRARY
Acc. No. A 54016
4 MAY 1978

ME-1977-M-JAI-NUM

CERTIFICATE

This is to certify that the thesis entitled "A Numerical Solution of the Inverse Problem for Hydraulic Turbomachines using a Singularity Method" by N.K. Jain is a record of work carried out under my supervision and has not been submitted elsewhere for a degree.

(Raminder Singh)
Assistant Professor
Dept. of Mech. Engg.
I.I.T., Kanpur

POST GRADUATE OFFICE
This thesis has been approved for the award of the Degree of Master of Technology (M. Tech.) in accordance with the regulations of the Indian Institute of Technology Kanpur
Dated. 1.12.77

CERTIFICATE

This is to certify that the thesis entitled "A Numerical Solution of the Inverse Problem for Hydraulic Turbomachines using a Singularity Method" by N.K. Jain is a record of work carried out under my supervision and has not been submitted elsewhere for a degree.

(Raminder Singh)
Assistant Professor
Dept. of Mech. Engg.
I.I.T., Kanpur

POST GRADUATE OFFICE
This thesis has been approved
for the award of the Degree of
Master of Technology (M. Tech.)
in accordance with the
regulations of the Indian
Institute of Technology Kanpur
Dated. 1-12-77

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to Dr. Raminder Singh, my supervisor, for his constant encouragement, invaluable guidance, comments and criticism throughout the tenure of this work.

I am thankful to Dr. M.M. Oberoi, Dr. M. Ramamohana Rao, Dr. V.K. Garg, Dr. K. Banerjee and Dr. H.S. Mani for many fruitful discussions.

I owe my thanks to Mr. R.K. Girdonia, Mr. K.K. Sahu, Mr. R.K. Shevgaonkar and all my friends for their timely help and kind cooperation.

Thanks are also due to the staff of Computer Centre, I.I.T. Kanpur for extending me full cooperation in the execution of computer programs.

The excellent typing by Mr. S.K. Tewari and Mr. G.L. Misra and ink-drafting by Mr. D.K. Mishra are appreciated.

N.K. JAIN

CONTENTS

			PAGE
LIST OF FIGURES			vi
NOMENCLATURE			vii
SYNOPSIS			x
CHAPTER I :		INTRODUCTION	1
CHAPTER II :		THE ANALYTICAL FORMULATION OF THE PROBLEM	6
	2.1:	Meridional Potential Flow Solution for a DUCT Flow	7
	2.2:	Velocity Potential for the Perturbation Velocity	10
	2.3:	Induced Velocities due to distribution of Singularities Placed on the Camber Line of Blade Sections.	13
	2.4:	Determination of the Strengths of Vortices, Sources.	17
	2.5:	Blade Profile Shape.	20
	2.6:	An Estimation of v_*.	23
CHAPTER III:		PROCEDURE FOR NUMERICAL SOLUTION OF THE INVERSE PROBLEM.	25
CHAPTER IV :		NUMERICAL TECHNIQUES USED AND THEIR SUITABILITY.	29
	4.1:	Solution of Simultaneous Linear Algebraic Equations.	29
	4.2:	Interpolation.	31
	4.3:	Differentiation.	32
	4.4:	Smoothing.	35
	4.5:	Integration.	35
	4.6:	Fourier Approximation, Discrete Domain.	36
	4.7:	Shape of Camber Line.	37
	4.8:	Length of Streamlines in meridional plane.	38
CHAPTER V :		COMPUTATIONAL DIFFICULTIES AND THEIR REMEDY.	39
	5.1:	Evaluation of Poisson Integrals.	39
	5.2:	Convergence of Solution for the Integral Equation.	41
	5.3:	Floating Point Overflow/Underflow.	43

		PAGE
CHAPTER VI :	RESULTS AND DISCUSSION.	45
REFERENCES :		48
FIGURES :		51
APPENDIX I :	MATRIX METHOD FOR THE MERIDIONAL POTENTIAL DUCT FLOW SOLUTION.	60
1 :	Finite Difference Approximation.	60
2 :	Grid Size and Shape.	62
3. :	Types of Boundary Points.	62
APPENDIX II :	INDUCED VELOCITIES DUE TO SINGULARITIES PLACED ON CAMBER LINE.	63
1 :	Induced Velocities due to Sources/Sinks.	63
2 :	Induced Velocities due to Vortices.	65
APPENDIX III:	CORRECTION FOR THE SMALL DEPARTURES FROM "FREE-VORTEX" MEAN FLOW.	68
APPENDIX IV:	THE COMPUTER PROGRAMS	71
1 :	Meridional Potential duct flow Solution.	71
2 :	The Inverse Problem.	74
APPENDIX V :	INPUT DATA FOR THE NUMERICAL EXAMPLE.	80
APPENDIX VI :	FORTRAN LISTING.	81

LIST OF FIGURES

FIG. NO.		PAGE
1.	The grid for the meridional solution.	51
	(a) Finite difference grid and arrangement of grid points.	
	(b) Types of boundary grid points.	
	(c) Point (i,j) and neighbours near boundary.	
2.	(a) Blade sections on a stream surface.	52
	(b) Coordinate system and velocity components.	
3.	Symbols for blade section.	53
	(a) Blade geometry.	
	(b) Blade thickness.	
4.	Rows of singularities in $\beta-\theta$ plane.	53
5.	Meridional projection of rotor and stream surfaces.	54
6.	Velocity distribution specified on stream surfaces.	55
7.	Comparison of camber line shapes.	56
8.	Comparison of blade thickness on first and second stream surfaces.	57
9.	Comparison of blade thickness on third, fourth and fifth stream surfaces.	58
10.	Effect of slight change in specified velocity distribution for first stream surface.	59

NOMENCLATURE

C	Velocity relative to a blade row
E	Induced velocity defined by eqn.(2.37)
F_o	Variable, equal to $\frac{f'_{2c}}{f_{2c}} - \frac{f'_{1c}}{f_{1c}}$
f	Variable, defined by eqn.(2.16)
G	Variable, defined by eqn.(2.11)
H	Total enthalpy
I,J	Functions defined by eqns. (2.18) to (2.21)
i,j	Index number for grid points
ℓ	Total length of camber line
m	Summation Index
N	Number of blades
Q	Volume flow rate
r	Radial coordinate
S	Length along any coordinate direction
T	Function defined by eqn. (2.26)
t	Half the thickness of blade
u	Absolute velocity in α-direction
v	Absolute velocity in β-direction
w	Absolute velocity in θ-direction
x	Variable, equal to $N\tau$
z	Axial coordinate

Greek Symbols

α	Equipotential line coordinate
β	Meridional streamline coordinate

Γ	Circulation.
γ	Vortex strength
δ	Dirac's delta function
ε	$\tan^{-1}$ (slope of the camber line)
ζ	Vorticity
η	Coordinate normal to the camber line
Θ	Angle, defined by eqn. (2.22)
θ	Circumferential coordinate
κ	Correction factor for v_*, defined by eqn.(2.50)
λ_m	Parameter equal to mN
ξ	Coordinate along the camber line
σ	Source strength
τ	Function of β, defined by eqn. (2.9)
Φ	Velocity Potential
ψ	Stream function
ω	Angular velocity of the rotor

Subscripts

o	At the singularity
1	Upstream of singularity
2	Downstream of singularity
+	Suction side
-	Pressure side
*	For flow with infinity large number of blades
c	On the camber line
G	Cascade influence
h	At hub of the turbomachine

i	Inlet edge
k	Blade surface
ℓ	Trailing edge of the blade section
m	Value corresponding to summation index m
p*	Axisymmetric potential flow
q	Value related to sources/sinks
t	At casing of the turbomachine
α,β,θ	In the direction of respective coordinates
γ	Value related to vortices
η	In the direction normal to the camber line
ξ	Along the camber line

Superscripts

~	Perturbation value
-	Value relative to a blade row
'	Derivative w.r.t. β

SYNOPSIS

A NUMERICAL SOLUTION OF THE INVERSE PROBLEM FOR HYDRAULIC TURBOMACHINES USING A SINGULARITY METHOD

N.K. JAIN

MASTER OF TECHNOLOGY
Department of Mechanical Engineering
Indian Institute of Technology, Kanpur.

In the present work a computer program and computational results are presented for the blade design of a hydraulic turbomachine, using a singularity method based on Kashiwabara's work. Blade shapes are obtained from the specified velocity distributions on blade surfaces and camber line lengths on each stream surface. The flow through the turbomachine is assumed to be incompressible and inviscid.

Fortran program listing and description of the computer programs are also given.

CHAPTER I

INTRODUCTION

A designer is basically concerned with two essential aerodynamic problems. First is the "Analysis problem" in which he is concerned with the accurate prediction of the design and off-design performance of a given system of blades. While in the "Inverse problem" the designer requires to know what cascade of blades will perform a prescribed duty. In hydraulic turbomachines for given hub and casing geometry, he is at liberty to choose some among the following; the position of inlet edge of the blade; camber line lengths at various stream surfaces of revolution; camber line shapes; velocity or pressure distributions on suction and pressure surfaces of blade sections on a number of stream surfaces; thickness distribution; depending upon the method used for the solution.

A theoretical approach to the aerodynamic design of annulus and blade cascade of turbomachinery presents a formidable problem. It involves the solution of partial differential equations of the fluid flow in complicated geometries. It is only with the help of large fast computers that a realistic attempt can be made at all. In its entirety such an attempt would involve the solution of the equations of the three-dimensional, unsteady, compressible, viscous flow through the complex of rotating and stationary blades and associated ducts of a turbomachine. The size and speed of the present

day computers is not adequate for this purpose. Nor is the state of our knowledge about turbulent, separated and secondary flows. Usually efforts have been concentrated on the formulation of meaningful combinations of two-dimensional models of the processes involved to model the real three-dimensional flow. All references to three-dimensional methods should hereafter be understood to mean such formulations. Numerical techniques are applied to solve the associated two-variable non-linear partial differential equations which result from these models. The ultimate goal is a series of computer programs through which an aerodynamic design can be formulated and its off-design performance accurately predicted.

The present numerical solution is a computer program for the design of blades of any single stage hydraulic turbomachine. The computer may be regarded as a simulator and is used in the same manner as an experimental rig of great flexibility in which the effect of changes in design parameters may be investigated rapidly. A design can be finalised with some certainty before actual manufacture and testing with their problems of high cost and time delays.

While designing the blades of any axial, radial or mixed flow turbomachines, it would be desirable to obtain blade shapes by prescribing velocity distributions along the blade surfaces so as to avoid boundary layer separation, cavitation and other problems.

Many formulations of the inverse method for two-dimensional and three-dimensional cascades are available. Stanitz[1] gives a

two-dimensional approximate method widely used for getting an approximate design of blade cascade excluding the nose and tail region of blade element profile. Wu[2] published a method in which the shape of the mean streamline between blade element profiles and velocity distri along this stream line are prescribed together with the blade spacing, and the resultant blade element profile and velocity distribution along it are determined. The method involves approximations that are accurate for high solidity cascades, but has no direct control over the velocity distribution along the profile surface. Stanitz and Sheldrake[3] discussed a two-dimensional design method prescribing surface velocity distributions as a function of arc length of camber line for high solidity cascades, but it can not determine the blade shape in the region of stagnation points and also it is not applicable to blade design of turbomachines in which meridional stream lines are curved. Wilkinson[4] developed an improved version of singularity methods developed by Martensen and by Jacob and Riegels. It needs initial blade configuration, chord lengths, mean flow angle and thickness distribution in addition to the pressure or velocity distributions on the suction surfaces. Moreover, velocity distributions on pressure surfaces can not be specified in this method. The method works only for the design of blades of large pitch/chord ratio and stagger angles of less than 60°. In Ref.(5) Wilkinson has given some guide lines for a three-dimensional design method, which is similar to the above and is a simple modification of the analysis method described there. Murugesan[6] has given an improved version

of Wilkinson's[4] method, which operates by the systematic modification of geometry of an arbitrary initial blade shape to satisfy the requirement of desired velocity distribution, but is limited to two-dimensional incompressible flow. Reilly[7] has given a quasi-three-dimensional singularity method which approximates the stream surfaces by a connected series of truncated cones. Murai's method[8] is a three-dimensional method in which circulation distribution is prescribed. It is restricted to thin blades. Kashiwabara[9] has given a three-dimensional singularity method for the design of blades of hydraulic turbomachines, which is an improvement over Murai's method[8]. This method appears to be very useful for hydraulic turbomachines of almost any kind, and thus is selected for the numerical solution presented here.

At this point, the basic question one can ask while starting the solution for an "Inverse problem" is how to specify an optimum velocity distribution on the sections of blade element surfaces. Of course, the answer is not very easy, but a number of investigations (10, 11, 12, 13, 14) have been made which attempt to give guide lines for the optimal velocity distributions for two-dimensional subsonic cascade problems. The optimization is with respect to the following considerations.

1. Minimization of boundary layer separation losses. These can be avoided by prescribed velocities that do not decelerate rapidly enough to cause separation.

2. Avoidance of cavitation in incompressible flow, and shock losses in compressible flow. Prescribed velocities should not exceed certain maximum values dictated by these phenomena.

3. For compressible flow the desired flow rate can be assured by prescribed velocities that do not result in "choked flow" conditions.

CHAPTER II

THE ANALYTICAL FORMULATION OF THE PROBLEM

In the present solution of the inverse problem, the flow through the turbomachine is assumed to be incompressible and inviscid. Further, since the flow far upstream is irrotational, it will remain irrotational everywhere with respect to an inertial frame (lab. frame).

First the meridional potential duct flow solution is found i.e. the flow without blades or blade forces. Stream surfaces, even with blades present, are assumed to be close to the stream surfaces of the potential duct flow without blades, defined by the hub-casing boundaries.

Blade shapes are found on these potential flow stream surfaces and the flow on such surfaces, even with blades present, is assumed to be a two-dimensional flow. Vortices, sources and sinks are used to replace the blade in these two-dimensional sheets, analogous to plane potential flow singularity methods for solving the inverse problem.

When the blades are present, the velocity potential inside the blade row is assumed to consist of two parts; one due to an infinitely large number of blades, and the other part, a perturbation velocity potential $\tilde{\Phi}$, which yields the effect of a finite number of blades. A partial differential equation for $\tilde{\Phi}$ is obtained and then converted into an ordinary differential equation by assuming a suitable form of

solution which is periodic in the circumferential direction θ. This ordinary linear differential equation is solved by well known WKB approximation.

The vorticity distribution representing the blade camber can be directly obtained from the prescribed velocity distribution, so the first approximation of the camber line shape can be determined. Using the $\bar{\Phi}$ solution, induced velocities due to the vortex distribution $\gamma(\xi)$ on the camber line can be calculated. At this point, it is possible to find the velocities induced on blade camber line, both along the camber line direction ξ and normal to it in η direction. Clearly the sum of velocities in the ξ-direction due to mean flow and the induced flow should be equal to the mean of specified velocity distribution on upper and lower blade surfaces at each point. In general, this condition can not be satisfied unless a source distribution $\sigma(\xi)$ is also located on the camber line. This is the basis for finding $\sigma(\xi)$. It can be shown that the introduction of $\sigma(\xi)$ changes the effective value of $\gamma(\xi)$, which in turn demands a recalculation of $\sigma(\xi)$. The method finds the converged values of $\sigma(\xi)$ and $\gamma(\xi)$ iteratively. Once these are known the blade thickness $t(\xi)$ and camber line shapes are readily found. Hence the blade profile is completely defined.

2.1 Meridional Potential Flow Solution for a Duct Flow.

Both the Analysis and the Inverse problems need the meridional potential flow solution (without blades or blade forces) as an input.

This solution gives a set of meridional streamlines and the velocities along them.

There are two methods in common usage for finding the potential flow solution, (1) the relaxation or matrix through flow method, and (2) the streamline curvature method.

The "relaxation or matrix through flow method" consists in solving the equations of motion in terms of the stream function. Finite difference techniques are used for the solution of the resulting partial differential equation. The flow region is covered by a fixed grid system formed by two families of straight lines or curves or a combination. At each grid point the finite difference equation is satisfied, resulting in a set of simultaneous algebraic equations which may be solved using iterative techniques or direct matrix inversion.

The alternative "Streamline curvature method" consists in solving the equations of motion in terms of several variables, the main one being the meridional potential flow velocity v_{p*}. In this method the entire passage is divided into a chosen number of streamlines and normals to them. Then iteratively these streamlines and normals are modified by satisfying the velocity gradient equation along the normals and continuity equation in each stream tube.

Both methods solve the same flow equations but differences in techniques introduce different operational constraints and difficulties. Davis and Millar[15] have given a systematic comparison of these two different methods for the calculation of potential flow

from the point of view of a user who wishes to select one for development or implementation.

Significant effort towards optimising the damping factor and the curvature calculations is required if the streamline curvature method is to be used operationally. For the present work the competing matrix method was considered preferable.

The partial differential equation for incompressible, axisymmetric, potential duct flow in terms of stream function ψ , is (Ref. 16)

$$\frac{\partial^2\psi}{\partial z^2} + \frac{\partial^2\psi}{\partial r^2} - \frac{1}{r}\frac{\partial\psi}{\partial r} = 0 \qquad (2.1)$$

For the solution of this equation boundary conditions are specified at the passage boundaries in terms of ψ . We desire to impose uniform flow far upstream and downstream, for which purpose the passage boundaries are suitably extended upstream and downstream, and the values of ψ associated with the uniform flow are taken. On the passage walls, of course, ψ takes constant values.

The version of the matrix method used in the present work involves the covering of the whole region with a fixed regular square grid system as shown in fig. (1a), and writing a finite difference approximation to the principal eqn. (2.1) at every interior grid point and grid points near curved boundaries. This gives one algebraic equation for every interior or boundary grid point in terms of ψ values at that point and the neighbouring points or boundary values. The resulting system of simultaneous equations is solved by the Guass-Seidel iterative scheme.

The method offers fast convergence. The scheme used yields second order accuracy and is inherently stable. This method is a powerful one and can probably be successfully adapted for the blade-to-blade analysis problem. [The details of the above method are given in Appendix I] .

2.2 Velocity Potential for the Perturbation Velocity

The flow through a blade row as shown in fig.2(a) is assumed to be incompressible and inviscid. Since the flow is irrotational everywhere, the velocity potential Φ , for the absolute velocity satisfies the Laplace's equation. In the orthogonal curvilinear coordinate system (α,β,θ) shown in fig. 2(b), Laplace's equation for Φ can be written as,

$$\left\{\left(\frac{\partial\alpha}{\partial r}\right)^2 + \left(\frac{\partial\alpha}{\partial z}\right)^2\right\}\frac{\partial^2\Phi}{\partial\alpha^2} + \left\{\left(\frac{\partial\beta}{\partial r}\right)^2 + \left(\frac{\partial\beta}{\partial z}\right)^2\right\}\frac{\partial^2\Phi}{\partial\beta^2}$$

$$+ \left(\frac{\partial^2\alpha}{\partial r^2} + \frac{1}{r}\frac{\partial\alpha}{\partial r} + \frac{\partial^2\alpha}{\partial z^2}\right)\frac{\partial\Phi}{\partial\alpha} + \left(\frac{\partial^2\beta}{\partial r^2} + \frac{1}{r}\frac{\partial\beta}{\partial r} + \frac{\partial^2\beta}{\partial z^2}\right)\frac{\partial\Phi}{\partial\beta} + \frac{1}{r^2}\frac{\partial^2\Phi}{\partial\theta^2} = 0 \qquad (2.2)$$

here α = constant are the streamlines of the duct flow and β =constant are equipotential lines of that flow. The defining equations of α , β are as follows, with S_α, S_β the lengths along the respective lines

$$d\alpha = r\, v_{p^*}\, dS_\alpha \qquad (2.3a)$$

$$d\beta = v_{p^*}\, dS_\beta \qquad (2.3b)$$

On rearranging terms of the above eqn. (2.2) and assuming that when the blade row is operating near design conditions, the velocity

u_*, in α-direction can be neglected as compared to the velocity v_* in the β-direction, we get (Ref. 9)

$$\frac{\partial^2 \Phi}{\partial \beta^2} + \frac{1}{r^2 v_{p*}^2} \frac{\partial^2 \Phi}{\partial \theta^2} = 0 \tag{2.4}$$

The potential Φ can be considered to be the sum of two parts, the velocity potential of the mean flow in the limiting case when the blade row consists of an infinitely large number of blades, and a perturbation velocity potential $\tilde{\Phi}$, which yields the effect of finite number of blades. The former will independently satisfy eqn. (2.4), hence the latter $\tilde{\Phi}$ must satisfy

$$\frac{\partial^2 \tilde{\Phi}}{\partial \beta^2} + \frac{1}{r^2 v_{p*}^2} \frac{\partial^2 \tilde{\Phi}}{\partial \theta^2} = 0 \tag{2.5}$$

Considering the periodicity in θ-direction, $\tilde{\Phi}$ can be expressed as

$$\tilde{\Phi} = B(\beta)\, e^{i\lambda_m \theta} \tag{2.6}$$

where $\lambda_m = mN$, $(m = 1,2,3,\ldots)$ and N is the number of blades. Substituting eqn. (2.6) into eqn. (2.5), an ordinary differential equation for B is obtained

$$\frac{d^2 B}{d\beta^2} - \frac{\lambda_m^2}{r^2 v_{p*}^2} B = 0 \tag{2.7}$$

According to Kashiwabara[9] if the number of blades N is relatively large, the "WKB method" may be used to integrate eqn. (2.7).

u_*, in α-direction can be neglected as compared to the velocity v_* in the β-direction, we get (Ref. 9)

$$\frac{\partial^2 \Phi}{\partial \beta^2} + \frac{1}{r^2 v_{p*}^2} \frac{\partial^2 \Phi}{\partial \theta^2} = 0 \tag{2.4}$$

The potential Φ can be considered to be the sum of two parts, the velocity potential of the mean flow in the limiting case when the blade row consists of an infinitely large number of blades, and a perturbation velocity potential $\tilde{\Phi}$, which yields the effect of finite number of blades. The former will independently satisfy eqn. (2.4), hence the latter $\tilde{\Phi}$ must satisfy

$$\frac{\partial^2 \tilde{\Phi}}{\partial \beta^2} + \frac{1}{r^2 v_{p*}^2} \frac{\partial^2 \tilde{\Phi}}{\partial \theta^2} = 0 \tag{2.5}$$

Considering the periodicity in θ-direction, $\tilde{\Phi}$ can be expressed as

$$\tilde{\Phi} = B(\beta)\, e^{i\lambda_m \theta} \tag{2.6}$$

where $\lambda_m = mN$, $(m = 1,2,3,\ldots)$ and N is the number of blades. Substituting eqn. (2.6) into eqn. (2.5), an ordinary differential equation for B is obtained

$$\frac{d^2 B}{d\beta^2} - \frac{\lambda_m^2}{r^2 v_{p*}^2} B = 0 \tag{2.7}$$

According to Kashiwabara[9] if the number of blades N is relatively large, the "WKB method" may be used to integrate eqn. (2.7).

The method gives an exact solution in the limiting case $\lambda_m \to \infty$. When λ_m is large, the accuracy of the solution will be sufficient for practical purposes (Ref. 17).

The WKB solution to eqn. (2.7) can be expressed as:

$$B = a_1 G_1 \exp(\lambda_m \tau) + a_2 G_2 \exp(-\lambda_m \tau) \tag{2.8}$$

where

$$\tau \equiv \int P^{1/2} \, d\beta \tag{2.9}$$

$$P \equiv \frac{1}{r^2 v_{p^*}^2} \tag{2.10}$$

$$G_{1,2} \equiv P^{-1/4} \exp\left[\pm \frac{A_1}{2\lambda_m} - \frac{K}{4\lambda_m^2} \pm \frac{A_2}{8\lambda_m^3} + O\left(\frac{1}{\lambda_m^4}\right) \right] \tag{2.11}$$

$$A_1 \equiv \int K P^{1/2} \, d\beta \tag{2.12}$$

$$A_2 \equiv P^{-1/2} K' - \int K^2 P^{1/2} \, d\beta \tag{2.13}$$

$$K \equiv (4PP'' - 5P'^2)/16P^3 \tag{2.14}$$

a_1, a_2 are constants. Subscripts 1,2 correspond to upper and lower signs respectively. Hereafter prime denotes differentiation with respect to β. Now the fundamental solution of eqn. (2.5) will become

$$\tilde{\phi}_m = \left(C_{m1} \frac{f_1}{f_{10}} + C_{m2} \frac{f_2}{f_{20}} \right) \exp\{ i\lambda_m (\theta - \theta_o) \} \tag{2.15}$$

where

$$f_{1,2}(\beta) = C_{m1,2} G_{1,2} \exp(\pm \lambda_m \tau) \tag{2.16}$$

The subscript o denotes the value at the point (β_o, θ_o) as shown in fig. 4, and C_{m1}, C_{m2} are constants. From eqn. (2.15) the general solution of $\tilde{\Phi}$ can be expressed as:

$$\tilde{\Phi} = \sum_{m=1}^{\infty} \tilde{\Phi}_m = \sum_{m=1}^{\infty} \left(C_{m1} \frac{f_1}{f_{10}} + C_{m2} \frac{f_2}{f_{20}} \right) \exp \{ i\lambda_m (\theta - \theta_o) \} \quad (2.17)$$

2.3 Induced Velocities due to Distribution of Singularities Placed on the Camber Line of Blade Sections

The introduction of a finite but sufficiently large number of blades is expected not to disturb the stream surfaces of the mean flow. In this situation the flows on the stream sheets can be considered two-dimensional, and the blade can be replaced by distributed sources, sinks and vortices placed on the camber line. The flows on the mean stream surfaces are however not independent of each other and we consider the relation between them in Section 2.6.

In the above paragraph, mean flow surfaces refered to are the meridional stream surfaces with an infinite number of blades present. These surfaces are not, however, known in advance. It is shown in section 2.6, that the difference between them and the meridional potential duct flow surfaces is small, under certain conditions. These conditions are assumed to hold for the present analysis. Thus instead of analysing the three-dimensional flow passing through blades, we assume that the three-dimensional flow is composed of stacked two-dimensional flows, within the mean stream surfaces.

The potential of induced velocities due to sources, sinks and vortices placed on the camber line of blade sections as shown in fig. 4, is symmetric about $\theta = \theta_o$ with the periodicity of $2\pi/N$, and tends to a constant value, say zero, as $\beta \to \pm \infty$. Consequently from eqn. (2.17) induced velocities can be obtained as given below.

2.3(a) Induced Velocities in ξ and η directions

Refering to fig. 3(a), induced velocities in ξ and η directions, which are, respectively along and normal to the blade camber line, can be expressed as, (Ref. 9). [A condensed derivation of the following expressions is given in Appendix-II]

$$\frac{\tilde{C}_{\eta q}}{v_{p^*}} = \int_{\beta_i}^{\beta_\ell} \sigma_o \, I_q \, \frac{d\beta_o}{v_{p^*o} \cos \varepsilon_o} \tag{2.18}$$

$$I_q \equiv - \frac{N}{\pi r_o v_{p^*o}} \sum_{m=1}^{\infty} \frac{1}{F_o} \{ \frac{f'}{f_o} \sin \varepsilon \cos \lambda_m(\theta - \theta_o) + \frac{1}{r v_{p^*}} \frac{f}{f_o} \lambda_m \cos \varepsilon \sin \lambda_m(\theta - \theta_o) \}$$

$$\frac{\tilde{C}_{\xi q}}{v_{p^*}} = \int_{\beta_i}^{\beta_\ell} \sigma_o \, J_q \, \frac{d\beta_o}{v_{p^*o}(x - x_o) \cos \varepsilon_o} \tag{2.19}$$

$$J_q \equiv \frac{N}{\pi r_o v_{p^*o}} (x - x_o) \sum_{m=1}^{\infty} \frac{1}{F_o} \{ \frac{f'}{f_o} \cos \varepsilon \cos \lambda_m(\theta - \theta_o) - \frac{1}{r v_{p^*}} \frac{f}{f_o} \lambda_m \sin \varepsilon \sin \lambda_m(\theta - \theta_o) \}$$

$$\frac{\tilde{C}_{\eta \gamma}}{v_{p^*}} = \int_{\beta_i}^{\beta_\ell} \gamma_o \, I_\gamma \, \frac{d\beta_o}{v_{p^*o} (x - x_o) \cos \varepsilon_o} \tag{2.20}$$

$$I_\gamma \equiv \frac{N}{\pi}(x-x_o)\sum_{m=1}^{\infty}\frac{f'_{10}}{f_{10}}\frac{f'_{20}}{f_{20}F_o}\left\{\frac{f'}{f'_o}\frac{\sin\varepsilon}{\lambda_m}\sin\lambda_m(\theta-\theta_o)\right.$$

$$\left.\frac{1}{rv_{p*}}\frac{f}{f'_o}\cos\varepsilon\cos\lambda_m(\theta-\theta_o)\right\}$$

$$\frac{\tilde{C}_{\xi\gamma}}{v_{p*}} = \int_{\beta_i}^{\beta_\ell}\gamma_o J_\gamma \frac{d\beta_o}{v_{p*}\cos\varepsilon_o} \qquad (2.21)$$

$$J_\gamma \equiv -\frac{N}{\pi}\sum_{m=1}^{\infty}\frac{f'_{10}f'_{20}}{f_{10}f_{20}F_o}\left\{\frac{f'}{f'_o}\frac{\cos\varepsilon}{\lambda_m}\sin\lambda_m(\theta-\theta_o)+\right.$$

$$\left.\frac{1}{rv_{p*}}\frac{f}{f'_o}\sin\varepsilon\cos\lambda_m(\theta-\theta_o)\right\}$$

Where $x = N\tau$, and f takes the form of f_1 or f_2 according as $\beta \leq \beta_o$ or $\beta \geq \beta_o$, i.e. as β is upstream or downstream of a singularity.

Now, if we define a new variable Θ by,

$$x - x_i = \frac{x_\ell - x_i}{2}(1 - \cos\Theta) \qquad (2.22)$$

then, eqns. (2.19) and (2.20) become Poisson integrals as given below,

$$\frac{\tilde{C}_{\xi qc}}{v_{p*}} = \frac{1}{N}\int_o^\pi J_{qc}\frac{\sigma_o r_o \sin\theta_o}{\cos\varepsilon_o}\frac{d\theta_o}{(\cos\theta_o - \cos\theta_c)} \qquad (2.23)$$

$$\frac{\tilde{C}_{\eta\gamma c}}{v_{p*}} = \frac{1}{N}\int_o^\pi I_{\gamma c}\frac{\gamma_o r_o \sin\theta_o}{\cos\varepsilon_o}\frac{d\theta_o}{(\cos\theta_o - \cos\theta_c)} \qquad (2.24)$$

Hereafter subscript c denotes values at the camber line.

The functions I_{qc}, J_{qc}, $I_{\gamma c}$, $J_{\gamma c}$ in eqns. (2.18) to (2.21) express the combined influence of all blade singularities, including those of the whole of the cascade. In the limiting case $(\beta_o \to \beta_c,\ \theta_o \to \theta_c)$, the influence of cascade vanishes, and the terms $J_{qc}/(x-x_o)$ and $I_{\gamma c}/(x-x_o)$ tend to infinity. The resulting singular integrals can be evaluated for principal values only. To facilitate doing so numerically a function T is introduced such that,

$$\lim_{\substack{\beta_o \to \beta_c \\ \theta_o \to \theta_c}} \frac{J_{qc}}{x-x_o} = \lim_{\substack{\beta_o \to \beta_c \\ \theta_o \to \theta_c}} T \tag{2.25a}$$

$$\lim_{\substack{\beta_o \to \beta_c \\ \theta_o \to \theta_c}} \frac{I_{\gamma c}}{x-x_o} = \lim_{\substack{\beta_o \to \beta_c \\ \theta_o \to \theta_c}} T \tag{2.25b}$$

where,

$$T \equiv \frac{N \cos \varepsilon}{2\pi r_o v_{p*o}(x-x_o)} \tag{2.26}$$

It is easily seen that T expresses the influence of a singularity situated on the camber line, close to the singularity, i.e. in a region where the induced velocity due to the singularity is the dominant contribution.

Eqns. (2.18) to (2.21) give the mean value of induced velocities on the camber line (β_c, θ_c). The induced velocities on both sides of camber line and close to it can be expressed as follows :

$$\frac{\tilde{C}_{\eta qc \pm o}}{V_{p*}} = \pm \frac{1}{2} \frac{\sigma_c}{V_{p*}} + \frac{\tilde{C}_{\eta qc}}{V_{p*}} \qquad (2.27)$$

$$\frac{\tilde{C}_{\xi \gamma c \pm o}}{V_{p*}} = \mp \frac{1}{2} \frac{\gamma_c}{V_{p*}} + \frac{\tilde{C}_{\xi \gamma c}}{V_{p*}} \qquad (2.28)$$

Where the + and - signs correspond to the suction and pressure sides of the camber line respectively, as shown in fig. 3(a).

2.4 Determination of the Strength of Vortices and Sources/Sinks.

Vortices :

A vortex sheet is equivalent to a discontinuity in the tangential velocity of a magnitude equal to the local strength of the vortex sheet. This fact is used to find the vortex sheet strength $\gamma(\xi)$ along the camber line. Since the blade thickness is finite, though small, some relationship is required to relate the surface speeds to the speed at the camber line of an infinitely thin blade. This is provided by the Riegel's relation (Ref. 16),

$$\bar{C}_{\xi, \pm K} = \bar{C}_{\xi, c \pm 0} \sqrt{1 + \left(\frac{dt}{d\xi}\right)^2} \qquad (2.29)$$

The superscript '-' denotes values relative to a blade and subscripts $\pm K$ denote the upper and lower surface of the blade section as shown in fig. 3(a). The vorticity distribution on the camber line is therefore,

$$\gamma(\xi) = (\bar{C}_{\xi, c-0} - \bar{C}_{\xi, c+0}) = (\bar{C}_{\xi, -K} - \bar{C}_{\xi, +K}) \sqrt{1 + \left(\frac{dt}{d\xi}\right)^2} \qquad (2.30)$$

The blade thickness $t(\xi)$ is not known in advance, but values of $t(\xi)$ depend upon $\sigma(\xi)$, and are obtained from the previous iteration in an iterative solution process.

Source/Sink Strength :

As already stated, these strengths are chosen to satisfy the condition that $\bar{C}_{\xi c}$ the mean surface velocities along the camber line have the same value when calculated in two ways,

(i) as the mean of specified blade surface speed, corrected for the blade thickness by eqn. (2.29) i.e.

$$\bar{C}_{\xi c} = \frac{1}{2}\left(\bar{C}_{\xi,c+0} + \bar{C}_{\xi,c-0}\right) = \frac{1}{2}\left(\bar{C}_{\xi,+K} + \bar{C}_{\xi,-K}\right)\sqrt{1+\left(\frac{dt}{d\xi}\right)^2} \qquad (2.31)$$

(ii) as the combined velocity due to the mean and induced flows

$$\bar{C}_{\xi c} = v_* \cos\varepsilon + (w_* - \omega r)\sin\varepsilon + \tilde{C}_{\xi\gamma c} + \tilde{C}_{\xi q c} \qquad (2.32)$$

where v_* and w_* are the β and θ components of the flow due to an infinite number of blades.

Using eqn. (2.3b) and figs. 2(b) and 3(a) we can write $d\xi$ as,

$$d\xi = \frac{d\beta}{v_{p*}\cos\varepsilon} \qquad (2.33)$$

So the eqn. (2.19) can be rewritten as,

$$\frac{\tilde{C}_{\xi q}}{v_{p*}} = \int_0^{\ell} \sigma_o \frac{J_q}{(x-x_o)}\, d\xi_o \qquad (2.34)$$

Using eqns. (2.25a) and (2.34), for convenience in evaluating $\tilde{C}_{\xi qc}$, we write

$$\frac{\tilde{C}_{\xi qc}}{V_{p*}} = \int_0^{\ell} \sigma_o \frac{J_{qc}}{(x-x_o)} \, d\xi_o = \int_0^{\ell} \sigma_o \left\{ \frac{J_{qc}}{(x-x_o)} - T \right\} d\xi_o + \int_0^{\ell} \sigma_o \, T \, d\xi_o$$

or

$$\tilde{C}_{\xi qc} = \tilde{C}_{\xi qc,G} + E$$

so $\tilde{C}_{\xi qc,G}$ is given by,

$$\frac{\tilde{C}_{\xi qc,G}}{V_{p*}} = \int_0^{\ell} \sigma_o \left\{ \frac{J_{qc}}{(x-x_o)} - T \right\} d\xi_o \qquad (2.35)$$

This can be seen to be, essentially, the induced velocity on the blade camber line, due to the members of the cascade excluding the blade concerned, or briefly the effect of the cascade. Similarly E is given by,

$$\frac{E}{V_{p*}} = \int_0^{\ell} \sigma_o \, T d\xi_o \qquad (2.36)$$

is seen to be, essentially, the induced velocity on the blade camber line due to singularities on itself.

Now, using eqns. (2.31) to (2.36), E can be expressed as,

$$E \equiv \frac{1}{2} \cdot \overline{1 + \left(\frac{dt}{d\xi}\right)^2} \left(\bar{C}_{\xi,+K} + \bar{C}_{\xi,-K} \right) - V_* \cos\varepsilon - (w_* - \omega r) \sin\varepsilon - \tilde{C}_{\xi\gamma c} - \tilde{C}_{\xi qc,G} \qquad (2.37)$$

When the strength of sources and sinks $\sigma(\xi)$ in the right hand side of eqn. (2.36) is assumed to be known, it becomes

Fredholm's integral equation of first kind. Using eqns. (2.22) and (2.26) along with eqns. (2.29) and (2.33), the eqn. (2.36) can be rewritten as given in Ref. (9),

$$\frac{E}{v_{p^*} \cos \varepsilon} = \frac{1}{2\pi} \int_0^{\pi} \sum_o \frac{\sin \theta_o \, d\theta_o}{(\cos \theta_o - \cos \theta)} \tag{2.38}$$

where $$\sum \equiv \frac{\sigma}{v_{p^*} \cos \varepsilon}$$

In order to get the closed blade profile, the following closure condition is to be satisfied,

$$\int_0^{\pi} \sum \sin \theta \, d\theta = 0 \tag{2.39}$$

The solution of a similar integral equation as eqn. (2.38), which satisfies eqn. (2.39) has been obtained by Truckenbrodt[18]. The solution of above equations is given on similar lines by Kashiwabara[9] as follows.

$$\sum = -\frac{2}{\pi \sin \theta} \int_0^{\pi} \frac{E_o}{v_{p^* o} \cos \varepsilon_o} \frac{\sin^2 \theta_o \, d\theta_o}{(\cos \theta_o - \cos \theta)} \tag{2.40}$$

here as E, given by eqn. (2.37), also contains the unknown $\sigma(\xi)$, so eqn. (2.40) must be solved by successive approximations.

2.5 Blade Profile Shape

Blade profile shape can be defined by shape of the camber line, and the thickness distribution along the camber line. These can be obtained as given below.

2.5(a) Camber Line Shape

If the angle of incidence at the inlet to a blade row is zero, it can be expected that the row will exert negligible upstream influence on the flow. Making this assumption, the velocity in θ-direction, w_* in the flow of an infinitely large number of blades has the following relation with the circulation $\Gamma(\xi)$ about the z-axis.

$$\Gamma(\xi) = 2\pi\, r(\xi)\, w_*(\xi) \tag{2.41}$$

with $\Gamma(\xi)$ defined by,

$$\Gamma(\xi) = N \int_0^{\xi} \gamma(\xi)\, d\xi \tag{2.42}$$

and an irrotational inlet flow is assumed.

Assuming that the relative flow runs along the camber line and cannot penetrate through it, i.e. the velocity in η-direction at the camber line is zero, we get

$$\bar{C}_{\eta c} = \bar{C}_{\eta *} + \tilde{C}_{\eta c} = -v_* \sin\varepsilon + (w_* - \omega r)\cos\varepsilon + \tilde{C}_{\eta \gamma c} + \tilde{C}_{\eta q c} = 0 \tag{2.43}$$

This equation will yield ε, which can be used for obtaining the θ-coordinate of the camber line as follows:

$$dS_\theta = r\,d\theta \quad , \quad dS_\beta = d\beta / v_{p*} \quad , \text{ hence}$$

$$\tan\varepsilon = dS_\theta / dS_\beta = \frac{r\, v_{p*}\, d\theta}{d\beta}$$

or $$\theta = \int_{\beta_i}^{\beta} \frac{\tan\varepsilon}{r\, v_{p*}}\, d\beta + \theta_{ci} \tag{2.44}$$

2.5(b) Thickness Distribution along the Camber Line

From continuity equation between half the thickness of the blade t and the distribution of sources and sinks, as shown in fig. 3(b), we get

$$\frac{1}{2} \int_0^{\xi} \sigma\, b\, d\xi = \int_0^{t} \bar{C}_{\xi}\, b\, d\eta \qquad (2.45)$$

where b is the distance between the mean stream surfaces. If t is assumed to be comparatively small, and using the velocity on the camber line, eqn. (2.45) can be approximated as follows,

$$\frac{1}{2} \int_0^{\xi} \sigma b\, d\xi \simeq \bar{C}_{\xi c}\, bt \qquad (2.46)$$

where $\bar{C}_{\xi c}$ is given by eqn. (2.32).

Since from fig. 2(b), $\frac{d\alpha}{dS_{\alpha}} = r v_{p^*}$, hence

$$dS_{\alpha} = b = \frac{d\alpha}{r v_{p^*}}$$

Because $d\alpha$ is constant between any two stream surfaces, eqn. (2.46) can be written as,

$$\frac{1}{2} \int_0^{\xi} \frac{\sigma}{r\, v_{p^*}}\, d\xi \simeq \frac{\bar{C}_{\xi c}}{r v_{p^*}}\, t \qquad (2.47)$$

This equation will give the thickness distribution along the camber line. In order to obtain closed blade profile the total strength of sources and sinks must be zero. Consequently the thickness at the trailing edge should be zero. So from eqn. (2.47) we get,

$$\int_0^{\ell} \frac{\sigma}{r v_{p*}} \, d\xi = 0 \qquad (2.48)$$

This is also a closure condition. The eqn. (2.39) is an alternative form of the above equation.

2.6 An Estimation of v_*

It is shown in Appendix-III that if the blade force in α-direction and $\frac{\partial}{\partial \alpha}(r w_*)$ are both negligible, then $v_* = v_{p*}$. The latter condition is seen to mean, on applying Euler's turbine equation along streamlines, that the work done on the fluid at different blade sections is so distributed that the total energy of the fluid along the β=constant lines is constant. This condition can be achieved in 'free-vortex' axial flow designs. In general it is not achieved and v_* differs from v_{p*}. For small departures from the free-vortex condition, and with the α-component of blade force negligible, Kashiwabara[9] gives the following relations:

$$\kappa = \int_{\alpha_h}^{\alpha} \frac{1}{v_{p*}^2} \left(\omega - \frac{w_*}{r}\right) \frac{\partial}{\partial \alpha}(r w_*) \, d\alpha + \kappa_h \qquad (2.49)$$

where $$\kappa = (v_* - v_{p*})/v_{p*} \qquad (2.50)$$

and κ_h is given by,

$$\kappa_h = -\frac{2\pi}{Q} \int_{\alpha_h}^{\alpha_t} \int_{\alpha_h}^{\alpha} \frac{1}{v_{p*}^2} \left(\omega - \frac{w_*}{r}\right) \frac{\partial}{\partial \alpha}(r w_*) \, d\alpha \, d\alpha \qquad (2.51)$$

where Q is the volume flow and has a relation,

$$\frac{Q}{2\pi} = (\alpha_t - \alpha_h) \tag{2.52}$$

Thus above equations can be used for calculating v_*, using known values of w_* and v_{p*}. [A derivation of above equations is given in Appendix-III].

CHAPTER III

PROCEDURE FOR NUMERICAL SOLUTION OF THE INVERSE PROBLEM

The procedure for the numerical solution of the inverse problem using a singularity method is given here. The analytical formulation and a brief outline of the procedure are already described in Chapter II.

Before proceeding with the solution, many parameters are to be selected beforehand such as; geometry of the flow passage of turbomachine i.e. of the hub and the casing; position of the inlet edge; design volume flow; number of blades and camber line lengths of blade sections on each stream surface. Velocity distributions are specified on both blade surfac as a function of camber line length. From the generalisation view point, all values are used in the non-dimensional form with respect to the inlet radius at the casing and blade speed at this position. For the numerical example presented here values of above parameters are given in Appendix-V.

In Fig. 5 the flow passage of a mixed flow turbomachine, five stream surfaces of axisymmetric potential duct flow and the blade inlet edge are shown, which are cited from Ref. (9). In the numerical example solved here, blade geometries are calculated with the prescribed velocity distributions shown in fig. 6, also cited from Ref.(9), for all the five stream surfaces considered.

The stepwise procedure for the numerical solution of the inverse problem, based on the theory given in Chapter II is as follows :

(1) The meridional potential duct flow solution without blades or blade forces for the given turbomachine is found using the method chosen in Section 2.1. This will give a set of streamlines and velocities along them, which are used as an input to the next part. The details of the procedure for finding the meridional potential flow solution using a matrix method are given in Section 2.1 and Appendix-I.

(2) Lengths of streamlines are calculated from the coordinates of streamlines, starting from the given position of the inlet edge.

(3) As is well known, a good specification of the blade geometry should contain relatively more data for the highly curved nose and tail regions than for the rest of the blade. Further, in the interests of limiting computer storage and running time demands, the blade geometry is sought to be specified at the fewest possible points - nineteen in the present case. Glauert's transformation $\xi/\ell = (1 - \cos\phi)/2$ yields a suitable distribution of points in physical space, when equal intervals are chosen for ϕ . The given table of values for the velocity distribution $\bar{C}_{\xi+K}(\xi)$ is transformed to table of values at the chosen ϕ points by interpolation.

(4) From this specified velocity distribution along blade surfaces the first approximation of vortex distribution along camber lines is calculated from eqn. (2.30), assuming zero blade thickness. As the circulation of the flow at inlet is zero, from eqns. (2.41) and (2.42) w_* can be calculated.

(5) Now assuming that v_* and v_{p*} are the same for the first approximati eqn. (2.43) is used for calculating the angle ε of the camber line

with S_β-direction, neglecting induced velocities due to vortices, sources and sinks. The camber line slope $\tan\varepsilon$, will be used to get the coordinate S_β of the camber line. The eqn. (2.44) will give the θ-coordinate of the camber line at corresponding points. These S_β and θ-coordinates are the first approximation to the shape of camber lines. These camber lines are taken as relative streamlines on respective stream surfaces.

(6) Variables appearing in the WKB solution τ, P, K, f_1 and f_2 are calculated using eqns. (2.9) to (2.14) and (2.16). Derivatives of these variables P', P'', K', f_1' and f_2' are calculated using suitable numerical techniques.

(7) Using the vortex distribution obtained above and variables along with their derivatives computed in (6), the induced velocities due to vortices placed on the camber line are calculated from eqns. (2.21) and (2.24). The infinite series appearing in these equations should be truncated suitably.

(8) The prescribed velocity distribution and eqn. (2.37) are used together to solve the integral eqn. (2.40). In the integral eqn.(2 the unknown $\sigma(\xi)$ appears on both sides. This integral equation mus therefore be solved by successive approximations. The solutions on different stream surfaces yield the source-sink distribution $\sigma(\xi)$ along the camber lines on respective stream surfaces.

(9) Induced velocities due to source-sink distribution $\sigma(\xi)$ are calculated using eqns. (2.18) and (2.23). Here also the infinite series appearing in these equations should be truncated suitably.

(10) Now the shape of the camber line is corrected according to the condition that the relative flow runs along the camber line. Eqns. (2.41) to (2.44) along with eqn. (2.30) are used for this purpose. This step is quite similar to step (5).

(11) From the source-sink distribution obtained in step (8), the blade profile thickness is calculated using eqns. (2.32) and (2.47).

(12) Now v_* is evaluated from known values of v_{p*} and w_* on each stream surface, as given in section 2.6, using eqns. (2.49) to (2.52).

(13) The vortex distribution is calculated again from eqn. (2.30) taking into account the blade thickness obtained in (11).

The stepwise procedure given above from step (6) to step (13) is iterated until the shape of the camber line and blade thickness converge with desired accuracy.

It is to be noted that step (12) given above is included in the iteration loop. The reason is, that w_* will change due to successive approximation of blade thickness. Hence the iterative calculation of v_* is also needed, because eqns. (2.49) to (2.51) contain w_* as one of the variables.

The procedure described above will remain the same whether the hydraulic turbomachine is a turbine or a pump, except that the angular velocity term ω in the analytical formulation will be $-\omega$ for a pump.

CHAPTER IV

NUMERICAL TECHNIQUES USED AND THEIR SUITABILITY

The adoption of numerical techniques for the solution of engineerin problems has got an important place. But considerable attention must be paid while using these techniques.

The use of numerical analysis is both a science and an art. It is a key to the specialist in the field but is often misunderstood by nonspecialists. Like most arts it is learned by practice. Principles there are, but even these remain unreal until actually applied.

A person wishing to solve an equation or some other problem numerically, usually has a wide choice of methods. Most problems of engineering interests are not so generously endowed, and for them the choice is restricted to a few methods. A reasonable balance must be kept between memory or storage requirements, accuracy, and speed of computatic

In the present numerical solution, the above considerations were kept in mind while selecting any particular numerical technique, by the limited knowledge gained by the author. The numerical techniques used along with their suitability for the present numerical solution are discussed below.

4.1 Solution of Simultaneous Linear Algebraic Equations :

In the meridional potential duct flow solution, a system of simultaneous linear algebraic equations results from the finite differe

approximation of the partial differential eqn. (2.1) at each grid point. The number of grid points is very large (about 2000 for the present case), for getting an accurate solution of eqn. (2.1) in the given flow region. It is almost impossible to store a coefficient matrix of this size (2000 × 2000) in any computer.

One important property of such a system of algebraic equations is that it has, at the most, only five non-zero terms in each equation, and the coefficient matrix contains a high proportion of zero elements. Consequently attempts to work entirely in terms of matrices and, in particular, to store all their elements is either impossible or extremely wasteful of computer storage. Rather, here the Gauss-Seidel iterative method[19] can be used by the repeated application of a single algebraic equation throughout the whole system of grid points. Using double space subscripts are shown in fig. 1(a), this amounts to computing a new approximation of $\psi_{i,j}$ at every interior grid point from a simple formula. For the present case, e.g.,

$$\psi_{i,j(new)} = \frac{\psi_{i-1,j} + \psi_{i+1,j} + \psi_{i,j-1} + \psi_{i\ j+1}}{4} - \frac{(\psi_{i,j+1} - \psi_{i,j-1})\ \Delta x}{8r} \qquad \ldots(4.1)$$

where Δx is the mesh size. For grid points near the boundaries similar equations are used, which are given in Appendix-I, along with other details.

4.1(a) Acceleration of Convergence :

For accelerating the convergence of the iterative process used for the solution of the system of algebraic equations arising from the

elliptic problem, the method of successive overrelaxation is used. It is again an iterative technique, in which an improved estimate of $\psi_{i,j}$ is computed by applying the following formula at every grid point.

$$\psi_{i,j(\text{Improved})} = \psi_{i,j} + R_f \left(\psi_{i,j(\text{new})} - \psi_{i,j}\right) \qquad (4.2)$$

where, R_f is an arbitrary parameter. For $R_f = 1$, the process is identical to the Gauss-Seidel method. However, for a choice of R_f in the range of $1 < R_f < 2$, the convergence is more rapid for elliptic problems. Finding an optimum value of the parameter, $R_{f\ opt.}$, is somewhat difficult, and is discussed in detail by Forsythe and Wasow[20]. For the present case it was found by trial, and its optimum value is approximately 1.8. It was observed that number of iterations required for a chosen degree of convergence falls very rapidly, when the parameter is in the immediate vicinity of $R_{f\ opt.}$, and it is generally better to over-estimate $R_{f\ opt.}$ than to under-estimate it.

4.2 Interpolation :

In the meridional potential duct flow solution, a second order interpolation is employed near the boundary grid points for evaluating the functional values near the boundaries. As there we want functional values in terms of the first and second partial derivatives at the boundary grid point, so a Taylor series around the boundary grid point is a suitable interpolating formula. (Ref. 21).

Everywhere else, for interpolation in terms of only functional values, a Lagrangian quadratic polynomial is used, because the points are unequally

spaced. Acton[21] suggests that the Aitken's interpolation is quite suitable for unequally spaced points. It is no doubt superior in the sense that in Lagrangian interpolation there is no indication of the accuracy, whereas in Aitken's interpolation there is. However the accuracy of Lagrangian interpolation differs usually in 5th or 6th significant digit only as compared to Aitken's interpolation. Therefore the Lagrangian quadratic polynomial is used here, because of its simplicity.

4.3 Differentiation :

The calculation of derivatives numerically is a hazardous operation, especially when dealing with low accuracy data. And even for high accuracy data, derivatives higher than first are likely to have considerable errors. The inherent difficulty in numerical differentiation is that it tends to magnify small discrepancies or errors in the approximating function used for representing the data. That is why numerical differentiation is often called a noise-magnification process (in contrast to integration which is called a smoothing process). Because of the tendency to magnify discrepancies or errors, numerical differentiation is usually avoided wherever possible.

If it must be used considerable care is required if serious errors are to be avoided.

A number of methods are available if the data is equally spaced. And for many methods it is quite possible to estimate an error bound for the derivative computed. But for unequally spaced data it is quite difficult, if not impossible, to get an idea of the error present in the derivatives.

In the present numerical solution, almost everywhere the data is unequally spaced, and the derivatives even of the order four are needed. Keeping this in view, the selection of the differentiating method was very carefully studied. It was found that no available numerical method can control sizable errors in higher order derivatives. So a variety of formulae were tested on a sine curve to select one with less error in the higher order derivatives, along with reasonable storage requirements and speed of computation. It was necessary to impose the last two constraints as the derivatives are to be calculated frequently, and the solution as a whole needs a lot of storage.

Wilkinson[22] made an attempt to compare many formulae for calculating derivatives numerically, particularly for second order derivatives, using equally spaced data, where he used a cosine curve for the above purpose. For unequally spaced data Wilkinson[5] suggested a method of "parametric differentiation". In this method, both the dependent variable y and the independent variable x are assumed to be functions of the point index number i, and then formulae for equally spaced points are used to differentiate both the variables w.r.t i. Formulae suggested by Wilkinson[5] are as given below :

$$\frac{dy}{dx} = \frac{dy/di}{dx/di} \tag{4.3}$$

$$\frac{d^2y}{dx^2} = \frac{\frac{dx}{di}\frac{d^2y}{di^2} - \frac{dy}{di}\frac{d^2x}{di^2}}{(dx/di)^3} \tag{4.4}$$

Even this 'parametric differentiation method', when used for third or higher order derivatives, was found to result in sizable errors in the

derivatives of the sine curve used for testing purposes. It is expected that if the variation in the spacing of data is small and smooth, the results will be somewhat better, as the errors appearing may be due to the lack of smoothness of variables with respect to i.

It was found that fitting a second order polynomial through three points to find the derivative at the mid-point is somewhat superior (i.e. the error incurred is comparatively smaller). As in the case of equally spaced data when second order polynomials are used, the error in derivatives at the endpoints is almost twice that at the midpoints. It was found that a simple two point linear differentiation formula for the end points gave a reduced error as compared to second order polynomial. As this combination gives better results for the derivatives of an analytically known function (sine curve), and since the storage requirements and speed of computation are reasonable, it was used throughout for the calculation of derivatives in the present case. Derivatives higher than first are found by successive differentiation.

The formulae used for computing derivatives are given below. These are very simple and can be derived very easily.

At the end points,

$$\left(\frac{dy}{dx}\right)_1 = (y_2 - y_1)/(x_2 - x_1) \qquad (4.5)$$

For the last point formula will be similar to (4.5).

For rest of the points,

$$\left(\frac{dy}{dx}\right)_M = 2a\, x_M + b \qquad (4.6)$$

where,

$$a = \frac{y_{M-1}}{h_1\, h_{12}} - \frac{y_M}{h_1\, h_2} + \frac{y_{M+1}}{h_2\, h_{12}} \qquad (4.7)$$

$$b = -\left[\frac{y_{M-1}(x_M + x_{M+1})}{h_1\, h_{12}} - \frac{y_M(x_{M-1} + x_{M+1})}{h_1\, h_2} + \frac{y_{M+1}(x_{M-1}+x_M)}{h_2\, h_{12}}\right] \qquad \ldots (4.8)$$

and $h_1 = x_M - x_{M-1}$, $h_2 = x_{M+1} - x_M$, $h_{12} = h_1 + h_2$.

4.4 Smoothing :

The meridional potential duct flow solution gives the streamline coordinates (z,r) and meridional velocities v_{p^*} along them. As in the present case, first and higher order derivatives of K and P (defined by eqns. (2.10) and (2.14), which are basically functions of r,z and v_{p^*}) will be used in subsequent calculations, it is better to use least square polynomials to smooth the data (r,z,v_{p^*}), before calculating K and P and differentiating them.

A five point smoothing formula adopted by Wilkinson[5] for smoothing the value at the midpoint is,

$$y_M = -0.00174\, y_{M-2} + 0.1875\, y_{M-1} + 0.62847\, y_M + 0.1875\, y_{M+1} - 0.00174\, y_{M+2} \qquad \ldots (4.9)$$

This formula is used for smoothing r,z and v_{p^*} for each set except two points on each side of the range, which remain unchanged.

4.5 Integration :

As the data is unequally spaced almost everywhere, so a composite integration formula, using a second degree interpolating polynomial passing

through three consecutive points is employed for integration on sub-intervals. The repeated application of a low-order formula is usually preferred to the single application of a high-order formula, partly because of the simplicity of the low-order formulas and partly because of computational difficulties.

The formula used degenerates to Simpson's rule, for equally spaced data. Its derivation is quite simple and straight-forward.

4.6 Fourier Approximation, Discrete Domain :

Poisson integrals appearing in eqns. (2.23), (2.24) and (2.40) are singular integrals, and the function inside the integrals is highly non-linear. Because of presence of singularity these integrals must be handled carefully.

To evaluate these integrals numerically as given in Chapter V, the discrete Fourier approximation is used to represent the integrand. For the present case, only the Fourier sine-series approximation for the half range is sufficient. For the discrete domain 0 to π, the formulae given below are based on Ref. (23,24). The function $g(\theta)$ can be expressed as,

$$g_M(\theta) = \sum_{n=1}^{NF} B_n \sin n\,\theta_M \qquad (4.10)$$

$$M = 1,2,\ldots,N+1$$

and

$$B_n = \frac{2}{N} \sum_{M=1}^{N-1} g_M(\theta) \sin n\,\theta_M \qquad (4.11)$$

$$n = 1,2,\ldots,NF$$

where (N+1) is the number of points at which function is known, NF is the number of terms of the Fourier series, less than N. Using (N-1) terms of the above Fourier series (4.10), the function can be represented exactly at all (N+1) points. But if smoothing is desired, one can use less number of terms.

4.7 Shape of the Camber Line :

While determining the shape of the camber line, the angle ϵ is to be calculated using eqn. (2.43), which is a transcendental equation for ϵ. This equation could be transformed rather easily to the form :

$$\sin^2 \epsilon + B \sin \epsilon + C = 0 \qquad (4.12)$$

where B,C are known functions of (β, θ). Thus ϵ can easily be found.

However an alternative approach making use of the perturbation principle, as given below was also used.

As the induced velocities $\tilde{C}_{\eta\gamma c}$ and $\tilde{C}_{\eta q c}$ are small compared to other terms of the eqn. (2.43). So an initial approximation ϵ_o can be obtained as,

$$\epsilon_o = \tan^{-1} \left(\frac{w_* - \omega r}{v_*} \right) \qquad (4.13)$$

Defining $$\epsilon = \epsilon_o + \delta \qquad (4.14)$$

and using the usual perturbation procedure to find δ from eqn.(2.43), one gets

$$\delta = \frac{\tilde{C}_{\eta\gamma c} + \tilde{C}_{\eta q c} + (w_* - \omega r) \cos \epsilon_o - v_* \sin \epsilon_o}{v_* \cos \epsilon_o + (w_* - \omega r) \sin \epsilon_o} \qquad (4.15)$$

The perturbation process can be applied repeatedly to get a converged value for ε . However, it was found that only a single shot is sufficient to give a good approximation for ε .

4.8 Length of Streamlines in Meridional Plane :

Length of each streamline in meridional plane is to be calculated, as it is used for the solution. It is calculated from the leading edge, using the formula,

$$S_{\beta_{j+1}} = S_{\beta_j} + \int_{Z_j}^{Z_{j+1}} \sqrt{1 + \left(\frac{dr}{dz}\right)^2}\, dz \qquad (4.16)$$

where (r,z) are coordinates of a streamline, and dr/dz is found by fitting a local quadratic through three points (j, j+1, j+2), which gives $r = Az^2 + Bz + C$, and $dr/dz = 2Az+B$. The eqn. (4.16) gives

$$S_{\beta_{j+1}} = S_{\beta_j} + \frac{1}{2A} \int_{H_j}^{H_{j+1}} \sqrt{1 + H^2}\, dH \qquad (4.17)$$

where $H = 2Az + B$, hence

$$S_{\beta_{j+1}} = S_{\beta_j} + \frac{1}{4A} \left[H\sqrt{1 + H^2} + \log_e \left(H + \sqrt{1 + H^2}\right) \right]_{H_j}^{H_{j+1}} \qquad (4.18)$$

If the value of A is very small i.e. the streamline is locally a straight line, the following formula will be used instead of eqn.(4.18), which is obtained from eqn. (4.16) as,

$$S_{\beta_{j+1}} = S_{\beta_j} + (Z_{j+1} - Z_j)\sqrt{1 + B^2} \qquad (4.19)$$

CHAPTER V

COMPUTATIONAL DIFFICULTIES AND THEIR REMEDY

As indicated in Chapter IV, the use of numerical analysis is both science and an art. The science and art juxtaposition is due to an uncertainty principle which often occurs in solving problems, namely, that to determine the best way to solve a problem may require the solution of the problem itself.

The art aspect of solving problems numerically arises mainly in two places, (i) in choosing the proper method, and (ii) in circumventing the main road-blocks that always seem to appear. This implies the need for anyone who wishes to use numerical techniques to develope experience and with it, hopefully, intuition.

Some of the computational difficulties encountered during the present numerical solution along with the techniques used to circumvent them are described below.

5.1 Evaluation of Poisson Integrals

The Poisson integrals appearing in eqns. (2.23), (2.24) and (2.40) are singular integrals. Also the function inside the integrals is highly non-linear. For evaluating these integrals, proper care is needed because of the presence of the singularity. The procedure adopted for this purpose is as follows :

The Poisson integral can be written as,

$$\int_0^{\pi} g(\theta_o) \frac{\sin \theta_o \, d\theta_o}{(\cos \theta_o - \cos \theta)} \tag{5.1}$$

For the present case, the function $g(\theta_o)$ is known at nineteen points, unequally spaced. Interpolation is used to get values of g at intervals of $\pi/18$, that is at nineteen equally spaced points. At both ends of the range of integration the value of $g(\theta_o)$ is zero {which is obvious from the form of $g(\theta_o)$ in eqns. (2.23), (2.24) and (2,40)}. It can therefore be represented by a half-range Fourier sine-series, in the discrete domain. The Fourier approximation of the function $g(\theta_o)$, over the discrete domain, 0 to π, is given by eqns. (4.10) and (4.11). Using these equations, the integral (5.1) will have the form,

$$\int_0^{\pi} \{ \sum_{n=1}^{NF} B_n \sin n \theta_o \} \frac{\sin \theta_o \, d\theta_o}{(\cos \theta_o - \cos \theta)} \tag{5.2}$$

In this integral the B_n's are constants, so it can be written as:

$$\sum_{n=1}^{NF} \{ B_n \int_0^{\pi} \frac{\sin n \theta_o \sin \theta_o}{(\cos \theta_o - \cos \theta)} d\theta_o \} \tag{5.3}$$

with

$$\sin n\theta \sin \theta = \frac{1}{2} \{ \cos[(n-1)\theta] - \cos[(n+1)\theta] \} \tag{5.4}$$

The integral becomes a sum of integrals of the form,

$$\int_0^{\pi} \frac{\cos n \theta_o}{(\cos \theta_o - \cos \theta)} \cdot d\theta_o \tag{5.5}$$

Now, this type of integral may be evaluated using the following result, derived by Glauert and given in Ref. (16),

$$I_n = \int_0^{\pi} \frac{\cos n \theta_o}{(\cos \theta_o - \cos \theta)} \, d\theta_o = \pi \frac{\sin n\theta}{\sin \theta} \qquad (5.6)$$

The recurrence relation of such integrals, given in Ref.(16) is,

$$I_{n+1} = 2 I_n \cos \theta - I_{n-1} \qquad (5.7)$$

where $I_o = 0$ and $I_1 = \pi$, this is obvious from eqn. (5.6). The constant coefficients B_n appearing in (5.3) can be evaluated using eqn. (4.11).

5.2 Convergence of Solution for the Integral Equation

The integral eqn. (2.40) has the unknown variable σ, both on the left hand side as well as inside the integral in the form of variable E_o. It must,therefore,be solved by successive approximations, which is an iterative process. A single iteration for the solution of this integral equation at one stream surface, takes about 4.5 seconds of computational time for the computer used, which is quite significant if a large number of iterations is needed to get a converged solution. So obviously, one is interested in accelerating the convergence of the solution. The tricks employed for doing so in the present work are described below.

The method of successive over relaxation for accelerating the convergence was found quite suitable. The formula for over-relaxation

given by eqn. (4.2) was used but the parameter R_f, i.e. relaxation factor, was not kept constant, as for the meridional potential duct flow solution. For getting optimum convergence rate R_f was varied from 1.55 to 1.75 (found by trial), depending upon the percentage change in the value of σ, from iteration to iteration at any point; i.e. if the change is large, R_f will be low or vice-versa.

With successive over-relaxation the convergence is quite fast for some but not all stream surfaces, if we start with initial approximation as zero source distribution. In particular much trouble was encountered with the hub streamline, which also happens to be the longest. It never became clear, inspite of much experimentation, why this streamline was so troublesome. There was,however, some evidence that convergence was more rapid on the shorter streamlines i.e. those for which the mesh points were relatively closely spaced. The procedure finally adopted was based on the expectation that the pattern of source/sink distribution would be the same for all stream surfaces, and that an initial distribution displaying such a pattern would be beneficial. So the source distribution of the fifth stream surface was obtained first and used as an initial approximation for the fourth stream surface and so on. It was observed that the convergence for the other stream surfaces was improved considerably using this procedure together with the successive over-relaxation discussed above.

It was found that the convergence can be improved further, if we use successive over-relaxation after three unaccelerated iterations to allow the solution to "stabilize" itself.

5.3 Floating Point Overflow/Underflow

While evaluating f_1, f_2 numerically, using eqn. (2.16), the exponents of the exponential term of the order of $\pm$ 200 and above appear. For the computer used for computation, the limit on the values of this is about $\pm$ 88.

On carefully looking at the equations, it becomes clear that whenever f_1 or f_2 or their derived variables appear, they are in quotient form and the expected values of such quotients are reasonable. Now from the computational view point eqn. (2.16) has the following form,

$$f_{1,2} = A \exp (B_{1,2}) \tag{5.8}$$

$$\text{hence } f'_{1,2} = (AB'_{1,2} + A') \exp (B_{1,2}) \tag{5.9}$$

Using eqns. (5.8) and (5.9), the different types of quotients to be used for the computation can be expressed as,

$$\frac{f'_o}{f} = \left[\frac{A_o B'_o + A'_o}{A}\right] \exp (B_o - B) \tag{5.10}$$

$$\frac{f'}{f} = B' + A'/A \tag{5.11}$$

$$\frac{f'_o}{f'} = \left[\frac{A_o B'_o + A'_o}{AB' + A'}\right] \exp (B_o - B) \tag{5.12}$$

$$\frac{f_o}{f} = \left(\frac{A_o}{A}\right) \exp (B_o - B) \tag{5.13}$$

$$\frac{f_o}{f'} = \left[\frac{A_o}{AB' + A'}\right] \exp (B_o - B) \qquad (5.14)$$

$$F_o = B'_2 - B'_1 \qquad (5.15)$$

In eqns. (5.10) to (5.14) subscripts 1 or 2 of f, f', f_o, f'_o, B, B', B_o and B'_o are omitted, as above equations are true for both subscripts.

While using the above equations only the values of B_1, B_2, B'_1, B'_2, A and A' are to be stored, and eqns. (5.10) to (5.15) are used whenever appropriate quotients occur.

Using this procedure it was found that the exponent term in the quotients is always -ve and is reasonably small, which makes the computation simple and reliable.

CHAPTER VI

RESULTS AND DISCUSSION

The numerical solution for the inverse problem given here is for a mixed flow turbomachine shown in fig. 5. The computer program prepared is explained in detail in Appendix - IV. Although written for a particular geometry, it can be readily modified to deal with any type of hydraulic turbomachine.

The turbomachine, velocity distributions on all five stream surfaces and other data are cited from Ref. (9). In fig.7, the camber line shapes obtained are compared with those given in Ref. (9). Slight disagreement is present. It should be mentioned that the camber line shapes obtained had "almost converged". Due to time limitation, the program was not run to achieve convergence within the specified tolerence.

The blade thicknesses obtained by the present numerical solution are compared with those given in Ref. (9), as shown in figs. 8 and 9. There are considerable differences in magnitude, but the trend of thickness distribution is similar. Much effort was spent in trying to locate the cause of this deviation, as described below, but without success. The iteration to iteration variation of blade thicknesses is given in figs. 8 and 9. It can be seen that thicknesses displayed a convergent trend. Time limitations rather than any program defect prevented the solution being carried to a higher degree of convergence.

The deviation in the results as compared to those given in Ref.(9), may be due to some of the reasons given below :

(1) Experimentation showed that the solution is very sensitive to values of input data, particularly that of velocity distributions. As an example of this, a slightly varied velocity distribution was input for the first stream surface. The results of camber line shape and thickness distribution for the first stream surface are given in fig. 10 for the two slightly different velocity distributions. The changes are notable. Clearly a precise agreement with the results of Ref. (9) could not be expected since input data were read off rather small graphs given in Ref. (9).

(2) Further, it was found that even 5 to 10% error in meridional potential duct flow velocity v_{p*}, is sufficient to make the solution non-convergent. v_{p*} is one of the important basic inputs, and is not given in Ref. (9). The meridional potential duct flow solution used as input was found as described in Appendix-I. There is reason to believe that on some points near the boundaries, v_{p*} values may change by 3 to 4% depending upon the detailed numerical method used for the solution. This could be another factor causing disagreement.

(3) For the results given in Ref. (9), Kashiwabara[9] used upto 30 terms in the series for the calculation of the perturbation velocities, whereas for the present numerical solution only 11 terms are used, because of computer storage limitations.

(4) Kashiwabara[9] has used 4 terms in the series for $G_{1,2}$ in eqn.(2.11). The third and fourth terms involve calculation of third and fourth derivatives of P, which can have sizable errors, and can make the solution meaningless. The numerical methods examined for evaluating

these higher derivatives gave unacceptable results. So in the present work only 2 terms in the WKB solution series were used. This difference in the number of terms used must clearly have affected the solution obtained.

(5) For the estimation of v_* from v_{p*} using eqn.(2.49), the integration is to be carried out along β = constant line in the meridional plane. In the present numerical example it is assumed that the points having the same index number at every stream surface are on the β = constant line. This assumption is not perfectly valid, which is clear from the fig. 5, where the leading edge is seen to be β = constant line, but not the trailing edge. Because numerically it is rather cumbersome to locate the β = constant lines and then evaluate the integral, so for simplicity, the assumption stated above was used with the expectation that the change in v_* from v_{p*} will be small. This assumption can affect the numerical solution to some extent.

Conclusions and Scope for Further Work :

The present method for the numerical solution of the inverse problem, was selected with the expectation that the method will be very useful. But the method as programmed in the present work, has turned out to posses some shortcomings. It is not however clear that it is inferior to other existing methods for the solution of the inverse problem.

The effect of incorporating the corrections indicated in the foregoing paragraphs needs to be studied very closely. There is also the need to continue seeking better methods for the solution of the inverse problem, and efforts for the same should be continued.

REFERENCES

1. Stanitz J.D.; "Approximate design method for high solidity blade elements in compressors and turbines".
NACA TN 2408, July 1951.

2. Wu Chung-Hua, and Brown Curtis A.; "A method of designing turbomachine blades with a desirable thickness distribution for compressible flow along an arbitrary stream filament of revolution".
NACA TN 2455, Sept. 1951.

3. Stanitz J.D., and Sheldrake L.J.; "Application of a channel design method to high solidity cascades and tests of an impulse cascade with 90° of turning".
NACA Report No. 1116, 1953.

4. Wilkinson D.H.; "A numerical solution of the analysis and design problems for the flow past one or more aerofoils or cascades".
ARC, R & M No. 3545, April 1967.

5. Wilkinson D.H.; "Calculation of blade-to-blade flow in a turbomachine by streamline curvature".
ARC, R & M No. 3704, Dec. 1970.

6. Murugesan K., and Railly J.W.; "Pure design method for aerofoils in cascade".
J. Mech. Engg. Sci., Vol. 11, No. 5, 1969, pp. 454-467.

7. Railly J.W.; "Design of mixed flow cascades by Ackert's method".
Engineer, Vol. 224, No. 5827, Sept. 1967, pp. 405-411.

8. Murai H.; "Theory for feathering blades of diagonal and axial flow water turbines or pumps".
Report, Inst. High Speed Mechanics, Vol. 20, 1968/1969, pp. 177-195.

9. Kashiwabara Y.; "Theory on blades of axial, mixed and radial turbomachines by inverse method".
Bull. JSME, Vol. 16, No. 92, Feb. 1973, pp. 272-281.

10. Walker, C.J.; "A family of surface velocity distribution of axial compressor blading and their theoretical performance".
J. Engg. for Power, ASME trans., April 1976, pp. 229-241.

11. Citavy J.; "Two dimensional compressor cascades with optimum velocity distribution over the blade".
J. Engg. for Power, ASME trans., Vol. 97, Series A, No. 1, Jan. 1975, pp. 101-110.

12. Spacek L.; "The optimum choice of velocity distribution for design of aerodynamic devices".
Proc. Inst. for Machine Research, Fluid mechanics in turbomachinery, Czechoslovak Academy of Sciences, Prague, 1958.

13. Růžička M.; "Choice of the optimum surface velocity in the design of flow devices".
Monegraphs & Memoranda No. 10, SVUSS (National Research Institute for Machine design, Bechovice), 1971.

14. Papailion K.D.; "Boundary layer optimisation for the design of high turning axial flow compressor blades".
ASME Paper No. 70-GT-88.

15. Davis W.R., and Millar D.A.; "A comparison of the matrix and streamline curvature methods of axial flow turbomachinery analysis, from users point of view".
J. Engg. for Power, Vol. 97A, No. 4, Oct. 1975, pp. 549-560.

16. Csanady G.T.; "Theory of Turbomachines".
McGraw Hill Book Company, 1964, pp. 196-228.

17. Powell J.L., and Crasemann B.; "Quantum Mechanics"
pp. 5-13, page 140.

18. Truckenbrodt E.; "Die Berechung der Profilform bei vorgegebener Geschwindigkeitsverteilung".
Ingeniur Archiv, vol. 19 (1951), p. 366.

19. Carnahan B., Luther H.A., and Wilkes J.O., "Applied Numerical methods".
John Wiley and Sons., Inc., New York, 1969.

20. Forsythe G.E., Wasow W.R., "Finite difference methods for partial differential equations".
Wiley, New York, 1960.

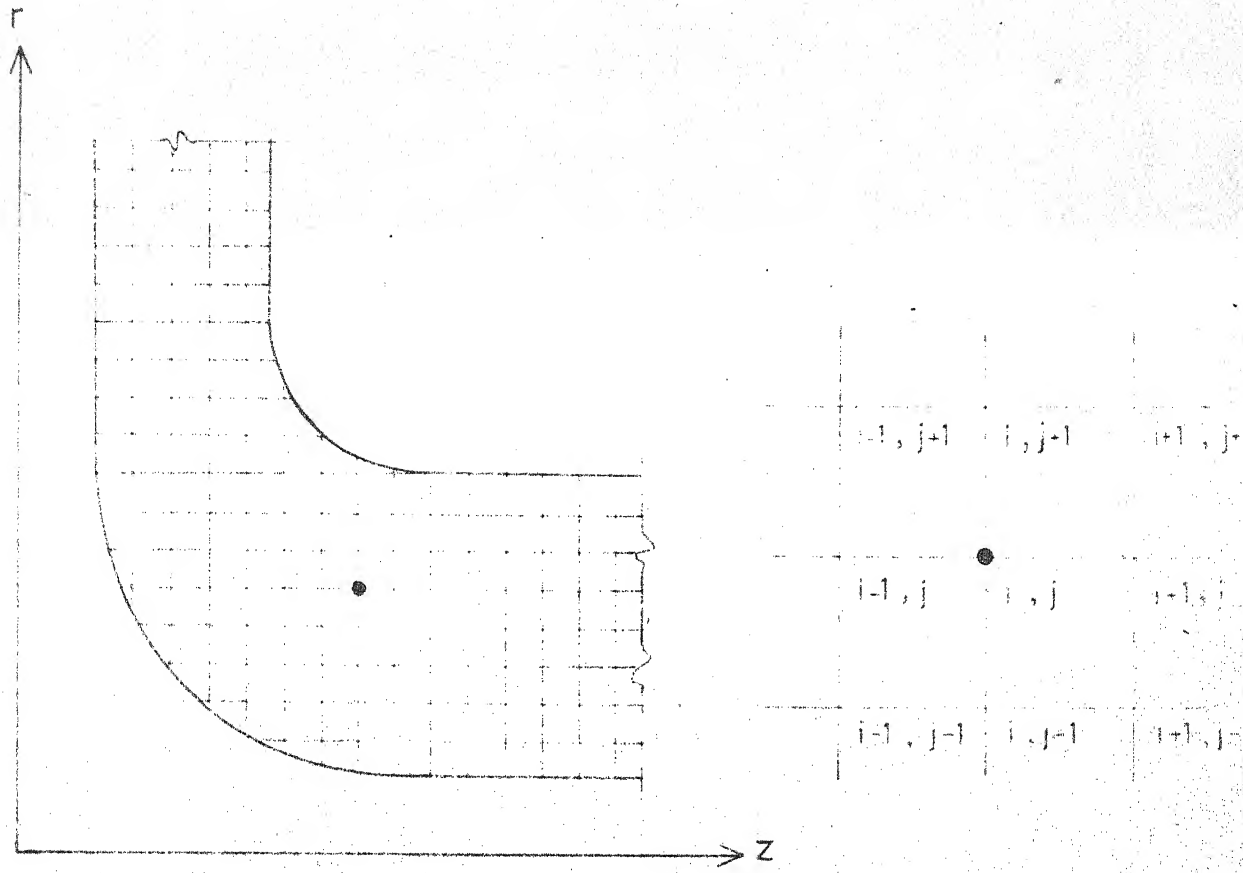
21. Acton F.S., "Numerical methods that work"
Harper and Row Publishers, New York, 1970.

22. Wilkinson D.H., "Stability, convergence and accuracy of two dimensional streamline curvature methods using quasi-orthogonals".
Inst. Mech. Engrs., Thermodynamics and Fluid mechanics convention, Paper 35, 1970.

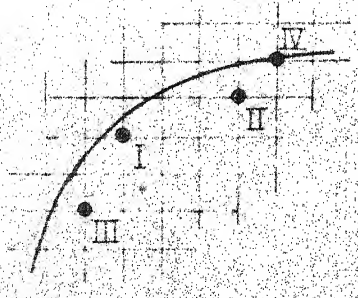
23. Ralston A., "A first course in numerical analysis".
McGraw Hill Book Company, New York, 1965.

24. Hildebrand F.B., "Introduction to Numerical Analysis". Tata McGraw Hill Publishing Co., Bombay, 1956.

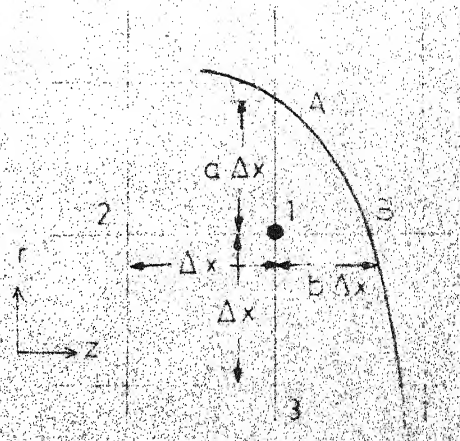
25. Marsh H., "Digital computer program for the through flow fluid mechanics in an arbitrary turbomachine using matrix method". ARC, R & M No. 3509, July 1966.



(a) Finite difference grid and arrangement of grid points.



(b) Types of boundary grid points



(c) Point (i,j) and neighbo near boundary.

Fig. 1 The grid for the meridional solution.

I.I.T. ...
CENTRAL ...
54001
Acc. No. ...

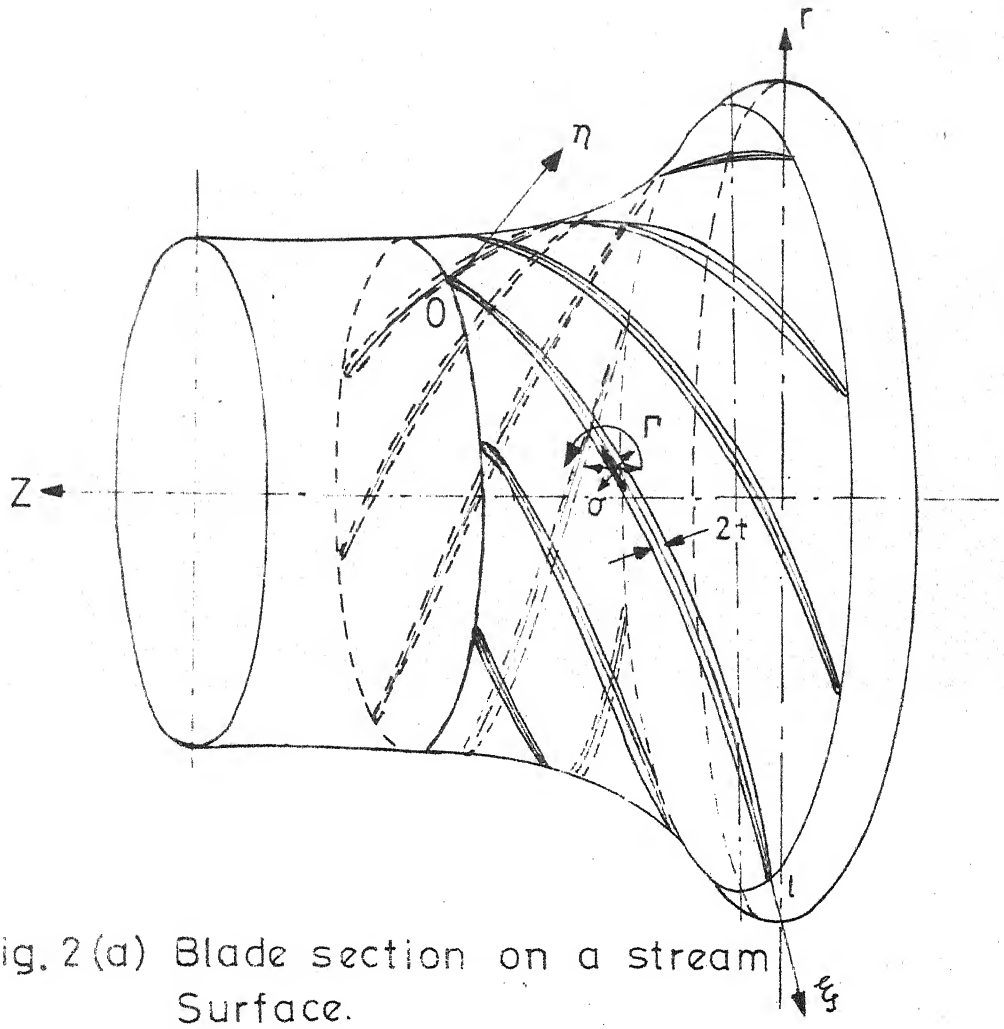


Fig. 2 (a) Blade section on a stream Surface.

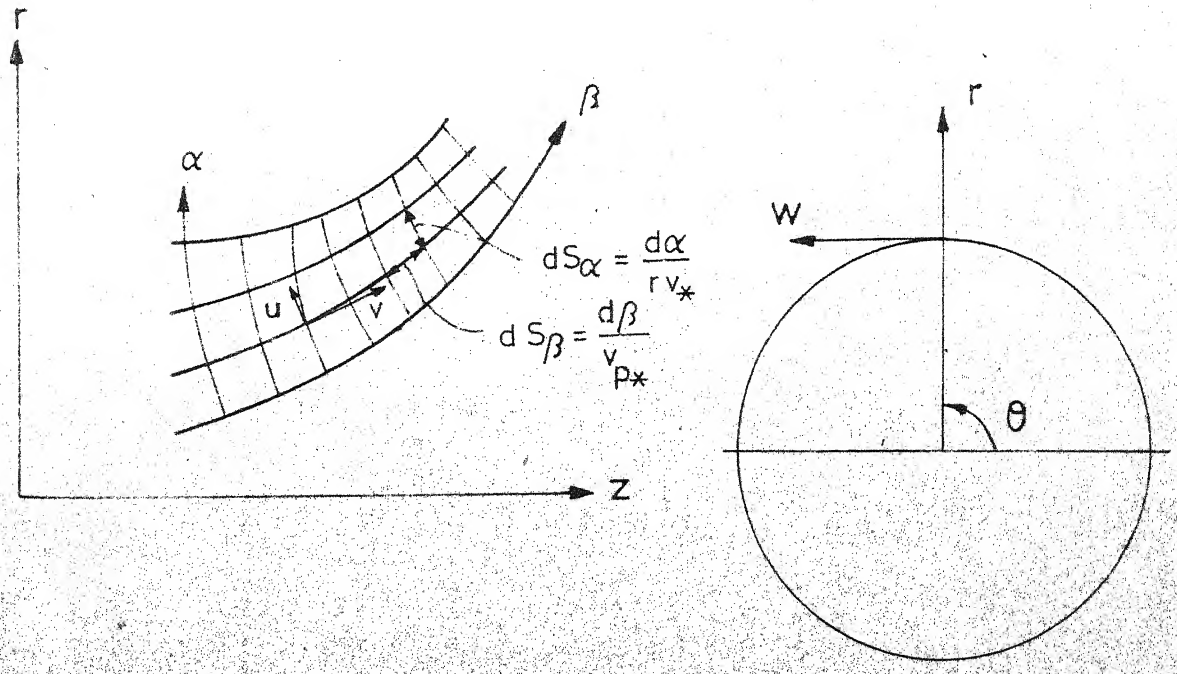
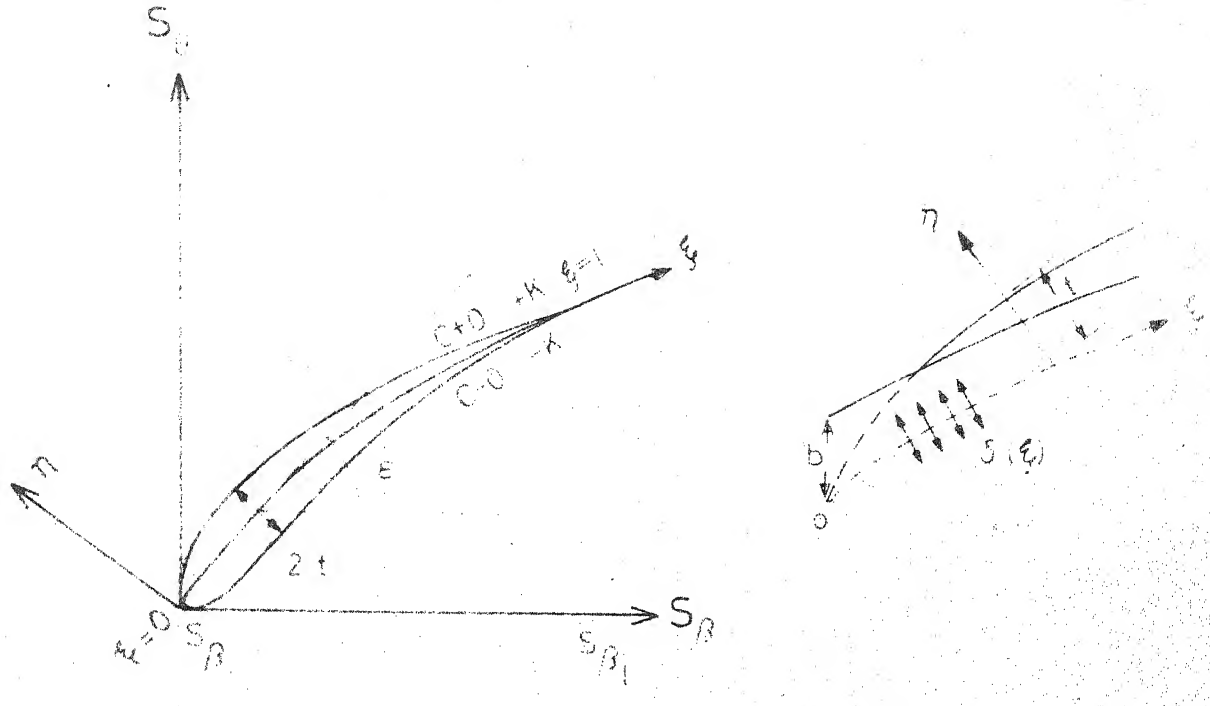


Fig 2 (b) Coordinate system and velocity components.



(a) Blade Geometry (b) Blade Thickness

Fig. 3 Symbols for blade section.

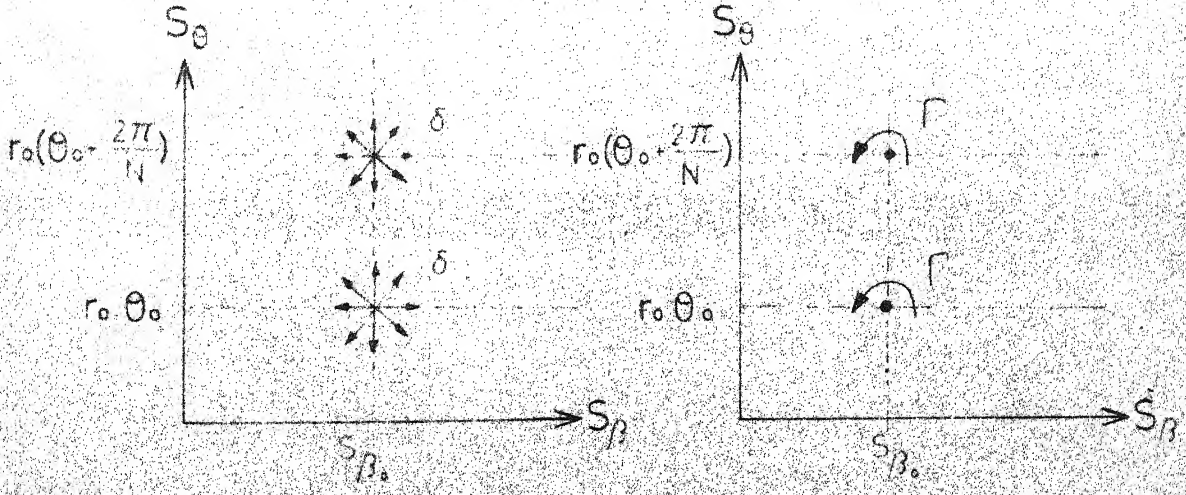


Fig. 4 Rows of singularities in β-θ plane.

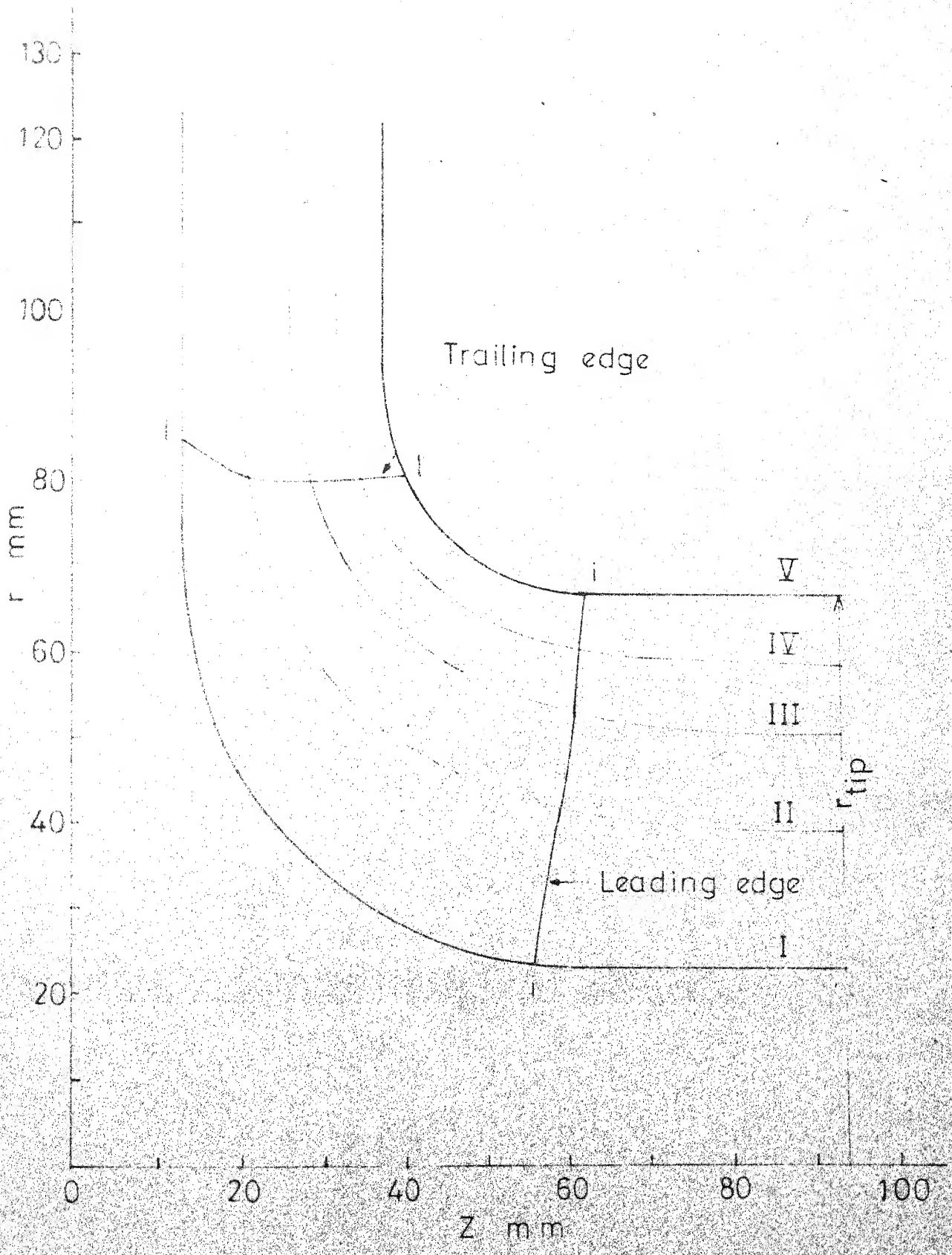


Fig. 5 Meridional projection of rotor and stream surfaces.

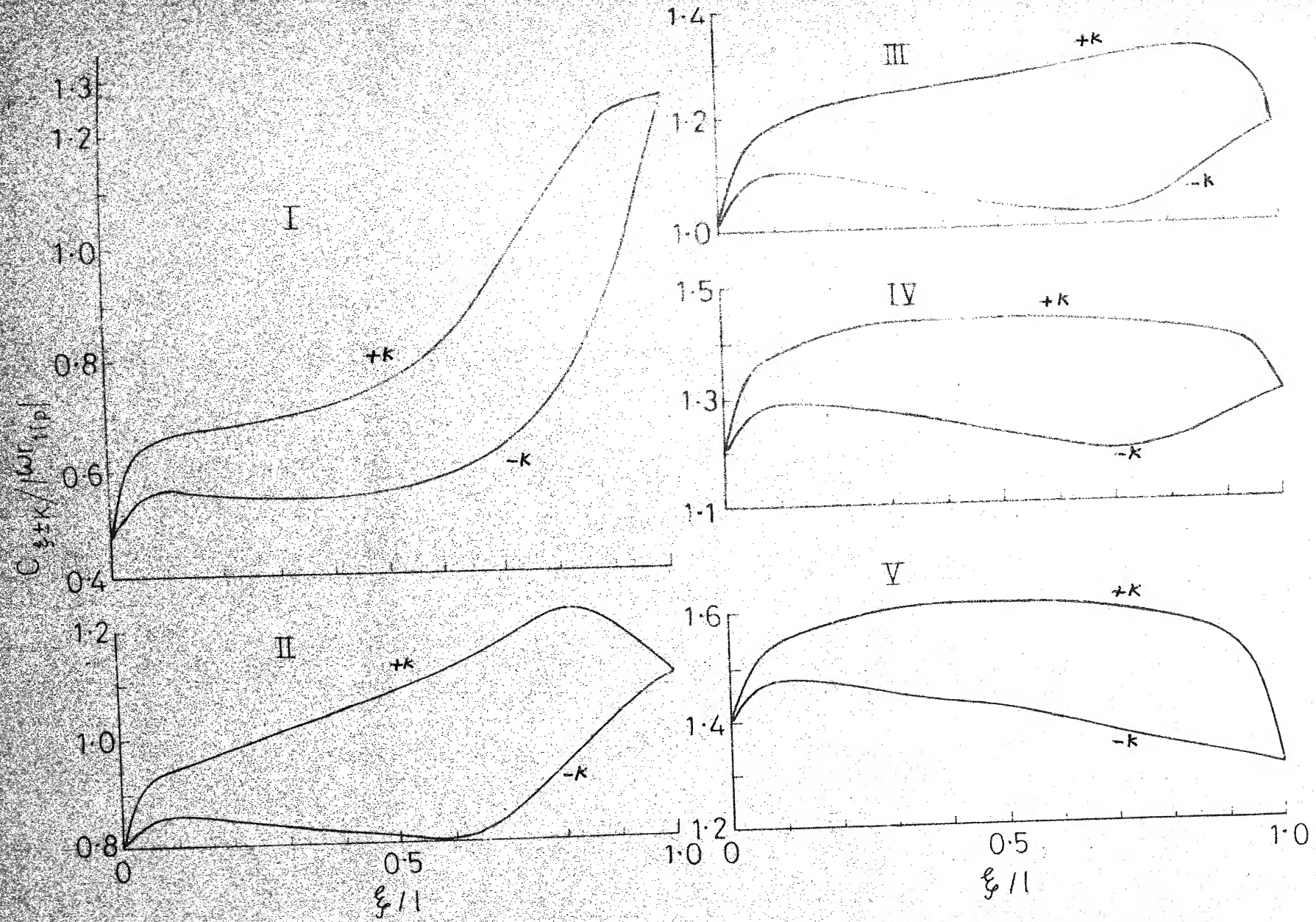


Fig. 6 Velocity distribution specified on stream surfaces.

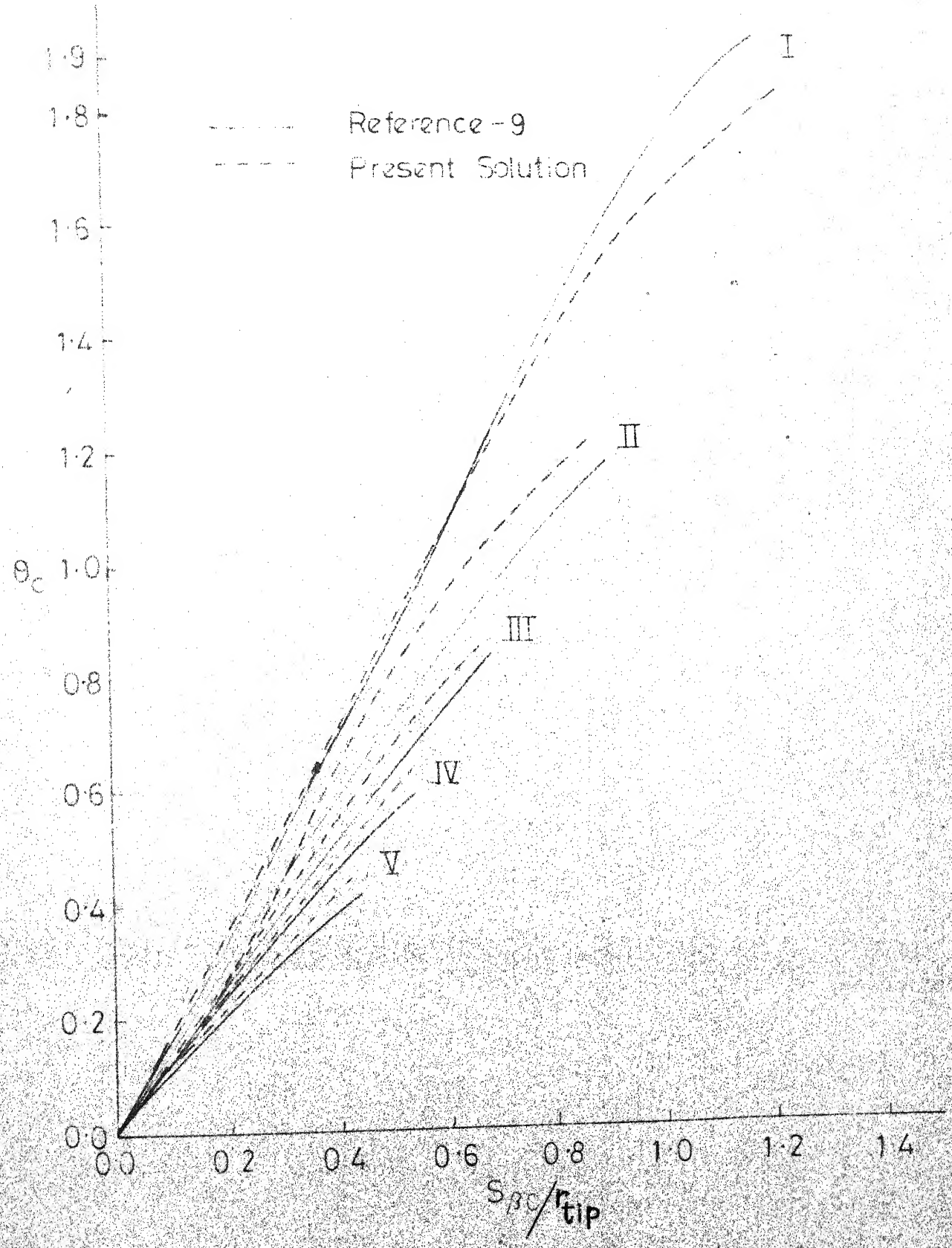


Fig. 7 Comparison of camber line shapes.

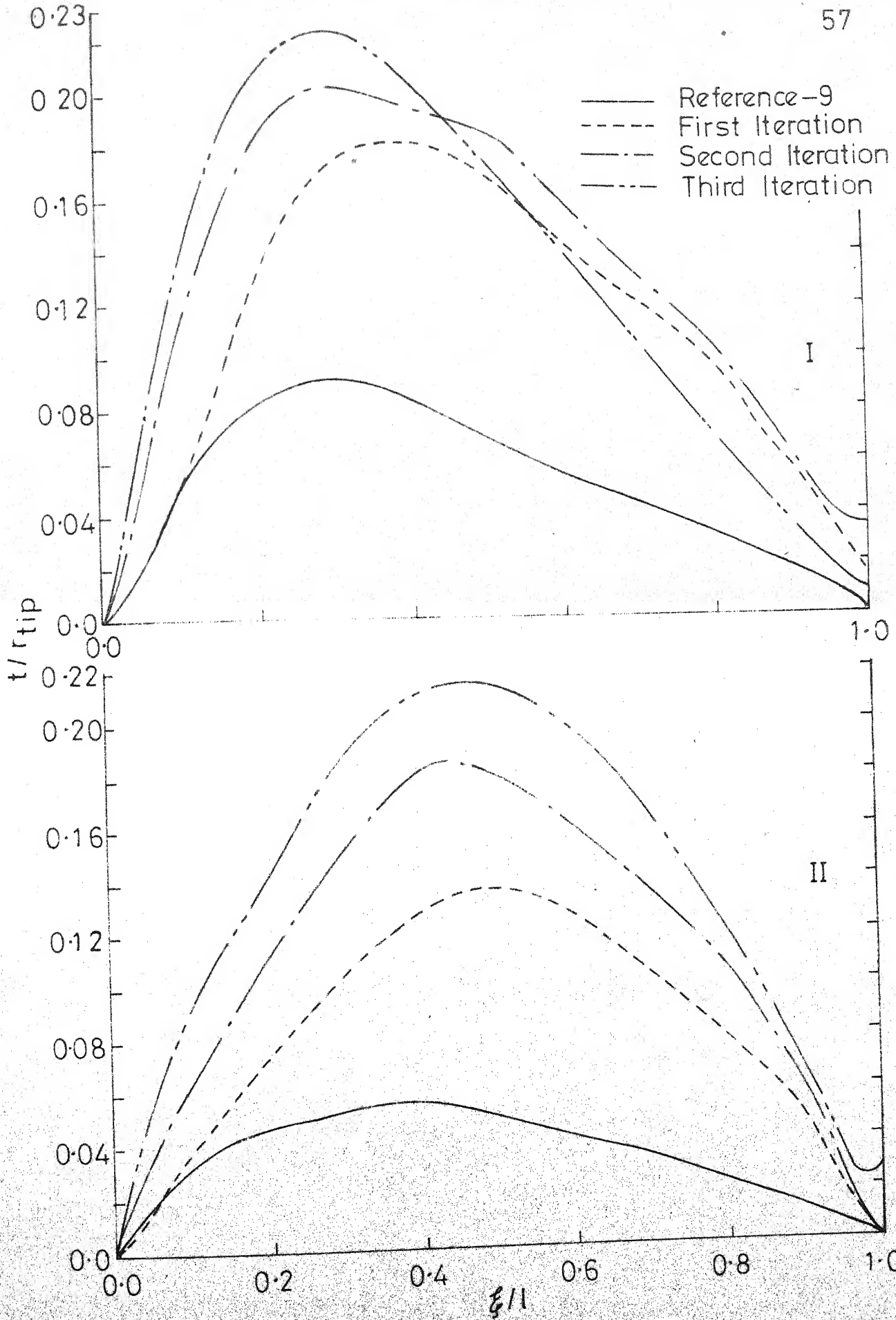


Fig.8 Comparis on of blade thickness on first and second stream surfaces.

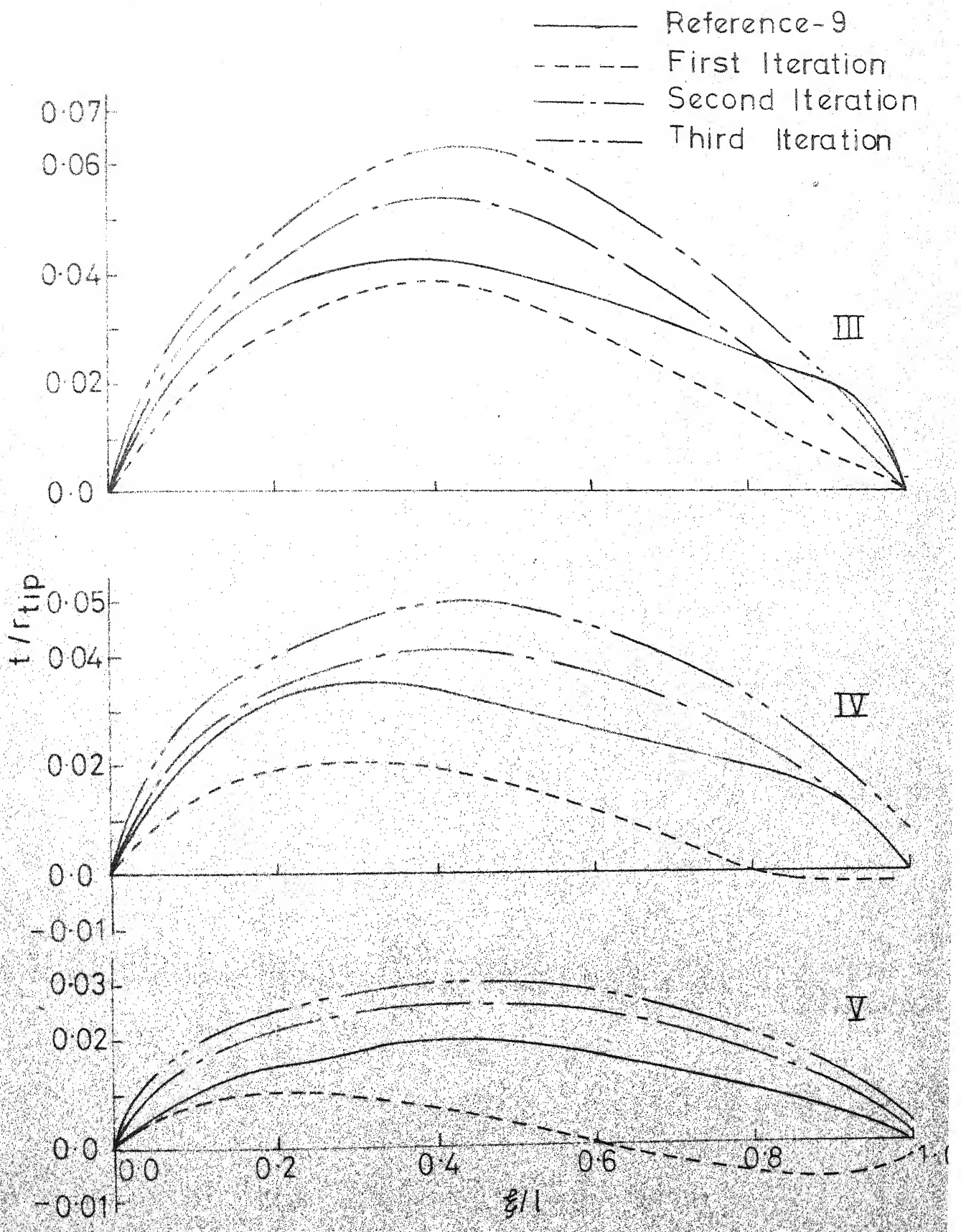


Fig. 9 Comparison of Blade thickness on Third, fourth and fifth stream surfaces.

APPENDIX - I

MATRIX METHOD FOR THE MERIDIONAL POTENTIAL DUCT FLOW SOLUTION

The matrix method involves covering of the region of interest with a fixed regular grid and writing a finite difference approximation to the principal eqn. (2.1) at every grid point as given below.

1. Finite-difference Approximation :

Marsh[25] used a curvilinear grid, because for the flow through a turbomachine of arbitrary shape, a quite suitable form of grid is a distorted or non-rectangular mesh, which is straight in one direction. He pointed out that while using curvilinear grid there is no difficulty in forming the finite difference approximation for derivatives in the direction of equally spaced points, but there is no general expression for the derivatives in the other direction and these are to be found using Taylor series approximation to get a required degree of accuracy. In the present numerical solution square grid is used since this leads to simple expressions for the interdependence of the functional values at neighbouring grid points.

At an interior grid point, using the notations shown in fig. 1(a), the finite difference approximation of principal eqn. (2.1) is given by,

$$\psi_{i,j} = \frac{\psi_{i-1,j} + \psi_{i+1,j} + \psi_{i,j-1} + \psi_{i,j+1}}{4} - \frac{(\psi_{i,j+1} - \psi_{i,j-1})\,\Delta x}{8r} \qquad (1)$$

where Δx is the mesh size.

Now for boundary grid points, consider a square grid shown in fig. 1(c), and let the boundary conditions be of the Dirichlet type. That is ψ is known along the curved boundary, of which we shall use points A and B. For simplicity, abandoning subscript notation temporarily, the relevant grid points are labelled as 1,2 and 3, as shown in fig. 1(c). For finding finite-difference approximation of eqn. (2.1) an appropriate Taylor's series expansion is used and unwanted derivatives are eliminated. Thus we have,

$$\psi_A = \psi_1 + a\,\Delta x \frac{\partial \psi}{\partial r} + \frac{(a\,\Delta x)^2}{2!} \frac{\partial^2 \psi}{\partial r^2} + 0\left[(\Delta x)^3\right] \tag{2}$$

$$\psi_B = \psi_1 + b\,\Delta x \frac{\partial \psi}{\partial z} + \frac{(b\,\Delta x)^2}{2!} \frac{\partial^2 \psi}{\partial z^2} + 0\left[(\Delta x)^3\right] \tag{3}$$

$$\psi_3 = \psi_1 - \Delta x \frac{\partial \psi}{\partial r} + \frac{(\Delta x)^2}{2!} \frac{\partial^2 \psi}{\partial r^2} + 0\left[(\Delta x)^3\right] \tag{4}$$

$$\psi_2 = \psi_1 - \Delta x \frac{\partial \psi}{\partial z} + \frac{(\Delta x)^2}{2!} \frac{\partial^2 \psi}{\partial z^2} + 0\left[(\Delta x)^3\right] \tag{5}$$

These equations enable us to find $\frac{\partial \psi}{\partial r}$, $\frac{\partial \psi}{\partial z}$, $\frac{\partial^2 \psi}{\partial r^2}$ and $\frac{\partial^2 \psi}{\partial z^2}$. The principal eqn. (2.1) then gives :

$$\psi_1 = C(D\psi_2 + E\psi_3 + F\,\psi_A + G\psi_B) \tag{6}$$

where, $C = \dfrac{1}{\frac{a+b}{ab} - \frac{(1-b)\,\Delta x}{2b\,r}}$, $D = \dfrac{1}{1+b} + \dfrac{b\,\Delta x}{2r(1+b)}$, $E = \dfrac{1}{1+a}$

$$F = \frac{1}{a(1+a)}, \quad G = \frac{1 - \frac{\Delta x}{2r}}{b(1+b)}$$

The method adopted for boundary grid points is a second order interpolation using Taylor's series. It was found that the above method is quite accurate for the cases analysed.

2. Grid Size and Shape :

While using curvilinear grid, Marsh[25] pointed out that it is difficult to give a general formula for the dependence of truncation error on the local shape of the grid. However this error will increase as the grid is distorted from a square shape. Similar is the case here, since the grid is not of square shape near the boundaries. Hence for such cases it is advisable to reduce the mesh size as the grid distortion increases. This will increase the amount of computer storage and computer time needed if a desired degree of accuracy is to be achieved.

3. Types of Boundary Points :

As shown in fig. 1(b), each point in row j can be placed into one of the following four categories :

Type I : Point $(i_{max,}\, j)$, if this is not of type III or IV.

Type II : Points (i,j), if any, for which $i_{max,\, j+1} > i > i_{max,\, j}$.

Type III : Point $(i_{max,}\, j)$, only if it is not of type IV and if $i_{max,\, j+1} = i_{max,j}$.

Type IV : Occurs if the curved boundary produces an intercept within $\pm\ \delta$ of a grid point, where $\delta << 1$.

Note that for any row j, points of types II, III and IV are mutually exculsive.

APPENDIX - II

INDUCED VELOCITIES DUE TO SINGULARITIES PLACED ON CAMBER LINE

1. Induced Velocities due to Sources/Sinks:

Velocity potential of the induced velocity due to a row of sources or sinks placed on the camber lines of N blade sections, as shown in fig. 4, is an even function about $\theta = \theta_o$, with the periodicity of $2\pi/N$, and tends to zero as $\beta \to \pm \infty$. Consequently from eqn. (2.17) it can be expressed as follows :

$$\left.\begin{aligned} \tilde{\Phi}_q &= \sum_{m=1}^{\infty} C_{m1} \frac{f_1}{f_{10}} \cos \lambda_m (\theta - \theta_o) \qquad \text{if } \beta \leq \beta_o \\ &= \sum_{m=1}^{\infty} C_{m2} \frac{f_2}{f_{20}} \cos \lambda_m (\theta - \theta_o) \qquad \text{if } \beta \geq \beta_o \end{aligned}\right\} \quad (1)$$

Where subscript q is used for values concerning sources/sinks, and subscripts 1,2 denote respectively the values upstream and downstream of a singularity. As $\beta = \beta_o$ is the matching plane between the upstream and downstream flow fields, the following continuity relations must be satisfied.

At $\beta = \beta_o \ (\theta \neq \theta_o)$

$$w_{q10} = w_{q20} \quad (2)$$

$$r_o (v_{q20} - v_{q10}) = q\,\delta(\theta - \theta_o) \quad (3)$$

where q is the strength of source or sink, and δ is the Dirac's delta function. Equation (3) embodies the velocity discontinuity at the source.

The Fourier expansion of a row of δ-functions can be obtained as,

$$\delta\left[\theta - \theta_o\right] = \frac{N}{2\pi} + \frac{N}{2\pi} \sum_{m=1}^{\infty} \cos \lambda_m (\theta - \theta_o) \tag{4}$$

Differentiating $\tilde{\Phi}_q$ w.r.t. θ will give r_{w_q}. Now using the condition given by eqn.(2), we get

$$C_{m1} \frac{f_1}{f_{10}} = C_{m2} \frac{f_2}{f_{20}}$$

as at $\beta = \beta_o$, $\frac{f_1}{f_{10}} = \frac{f_2}{f_{20}} = 1$, we get

$$C_{m1} = C_{m2} \tag{5}$$

Similarly differentiating $\tilde{\Phi}_q$ w.r.t. β will give v_q and then using the condition given by eqn. (3) along with eqns. (4) and (5), Kashiwabara[9] gets

$$C_{m1} = C_{m2} = \frac{\sigma_o}{r_o \, \Psi_{p*o}} \frac{N}{\pi} \frac{1}{F_o} \tag{6}$$

where $F_o = \frac{f'_{20}}{f_{20}} - \frac{f'_{10}}{f_{10}}$

An anomaly in Kashiwabara's formulation of the problem is to be noted here. V_q as obtained from (1) is a sine series in θ, with a zero mean value when averaged for constant β. A row of sources on the other hand necessarily gives rise to a flow with a non-zero mean value, as is evident from the first term in(4), and thus can not be represented by a potential of the form (1). Kashiwabara's formulation can probably be corrected in this respect. However, the limited initial objective of the present work was to obtain agreement with Kashiwabara's results

and this anomaly was ignored. Equations (5) to (8) are exactly those given in Ref.(9), where their derivation is not given.

Using constants C_{m1}, C_{m2} given by eqn. (6), we get

$$r\,\tilde{w}_q = \frac{\partial \tilde{\Phi}_q}{\partial \theta} = -\frac{N}{\pi}\int_0^{\xi} \frac{\sigma_o}{r_o\, v_{p*o}} \sum_{m=1}^{\infty} \frac{1}{F_o}\frac{f_2}{f_{20}}\lambda_m \sin\lambda_m(\theta-\theta_o)\, d\xi_o -$$

$$\frac{N}{\pi}\int_{\xi}^{\ell} \frac{\sigma_o}{r_o\, v_{p*o}} \sum_{m=1}^{\infty} \frac{1}{F_o}\frac{f_1}{f_{10}}\lambda_m \sin\lambda_m(\theta-\theta_o)\, d\xi_o \qquad (7)$$

and $$\frac{\tilde{v}_q}{v_{p*}} = \frac{N}{\pi}\int_0^{\xi} \frac{\sigma_o}{r_o\, v_{p*o}} \sum_{m=1}^{\infty} \frac{1}{F_o}\frac{f_2'}{f_{20}} \cos\lambda_m(\theta-\theta_o)\, d\xi_o +$$

$$\int_{\xi}^{\ell} \frac{\sigma_o}{r_o\, v_{p*o}} \sum_{m=1}^{\infty} \frac{1}{F_o}\frac{f_1'}{f_{10}} \cos\lambda_m(\theta-\theta_o)\, d\xi_o \qquad (8)$$

where ξ is the length along the camber line from the leading edge as shown in fig. 3(a).

2. Induced Velocity due to Vortices :

The potential of the induced velocity due to a row of vortices placed on the camber lines at shown in fig. 4, is an odd function about $\theta = \theta_o$ and tends to zero as $\beta \to \pm\infty$, so from eqn. (2.17) it can be expressed as follows.

$$\left.\begin{aligned} \tilde{\Phi}_\gamma &= \sum_{m=1}^{\infty} C_{m1}\frac{f_1}{f_{10}}\sin\lambda_m(\theta-\theta_o) \quad \text{if } \beta \le \beta_o \\ &= \sum_{m=1}^{\infty} C_{m2}\frac{f_2}{f_{20}}\sin\lambda_m(\theta-\theta_o) \quad \text{if } \beta \ge \beta_o \end{aligned}\right\} \qquad (9)$$

Where the subscript γ denotes the value concerning vortices. The following continuity condition must be satisfied for a row of vortices of strength Γ at the matching plane $\beta = \beta_o$.

At $\beta = \beta_o \ (\theta \neq \theta_o)$

$$v_{\gamma 10} = v_{\gamma 20} \tag{10}$$

$$r_o(w_{\gamma 20} - w_{\gamma 10}) = \Gamma \delta(\theta - \theta_o) \tag{11}$$

In the similar manner as described for sources/sinks, the constants C_{m1}, C_{m2} can be obtained using conditions (10) and (11) as,

$$C_{m1} = \frac{\gamma_o N}{\lambda_m \pi} \frac{1}{\{\frac{f'_{10}}{f'_{20}} \frac{f_{20}}{f_{10}} - 1\}} = \frac{-\gamma_o N}{\lambda_m \pi} \frac{f'_{20}/f_{20}}{F_o} \tag{12}$$

and $$C_{m2} = \frac{\gamma_o N}{\lambda_m \pi} \frac{1}{\{1 - \frac{f'_{20}}{f'_{10}} \frac{f_{10}}{f_{20}}\}} = \frac{-\gamma_o N}{\lambda_m \pi} \frac{f'_{10}/f_{10}}{F_o} \tag{13}$$

Using constants C_{m1}, C_{m2} given by eqn. (12) and (13), we get

$$r \tilde{w}_\gamma = \frac{\partial \tilde{\Phi}_\gamma}{\partial \theta} = -\frac{N}{\pi} \int_o^\xi \gamma_o \sum_{m=1}^\infty \frac{1}{F_o} \frac{f'_{10}}{f_{10}} \frac{f_2}{f_{20}} \cos \lambda_m (\theta - \theta_o) \, d\xi_o -$$

$$\frac{N}{\pi} \int_\xi^\ell \gamma_o \sum_{m=1}^\infty \frac{1}{F_o} \frac{f'_{20}}{f_{20}} \frac{f_1}{f_{10}} \cos \lambda_m (\theta - \theta_o) \, d\xi_o \tag{14}$$

and $$\frac{\tilde{v}_\gamma}{v_{p*}} = -\frac{N}{\pi} \int_o^\xi \gamma_o \sum_{m=1}^\infty \frac{1}{\lambda_m F_o} \frac{f'_{10}}{f_{10}} \frac{f'_2}{f_{20}} \sin \lambda_m (\theta - \theta_o) \, d\xi_o -$$

$$\frac{N}{\pi} \int_\xi^\ell \gamma_o \sum_{m=1}^\infty \frac{1}{\lambda_m F_o} \frac{f'_{20}}{f_{20}} \frac{f'_1}{f_{10}} \sin \lambda_m (\theta - \theta_o) \, d\xi_o \tag{15}$$

Where the subscript γ denotes the value concerning vortices. The following continuity condition must be satisfied for a row of vortices of strength Γ at the matching plane $\beta = \beta_o$.

At $\beta = \beta_o \ (\theta \neq \theta_o)$

$$v_{\gamma 10} = v_{\gamma 20} \tag{10}$$

$$r_o(w_{\gamma 20} - w_{\gamma 10}) = \Gamma \delta(\theta - \theta_o) \tag{11}$$

In the similar manner as described for sources/sinks, the constants C_{m1}, C_{m2} can be obtained using conditions (10) and (11) as,

$$C_{m1} = \frac{\gamma_o N}{\lambda_m \pi} \frac{1}{\{\frac{f'_{10}}{f'_{20}} \frac{f_{20}}{f_{10}} - 1\}} = \frac{-\gamma_o N}{\lambda_m \pi} \frac{f'_{20}/f_{20}}{F_o} \tag{12}$$

and $$C_{m2} = \frac{\gamma_o N}{\lambda_m \pi} \frac{1}{\{1 - \frac{f'_{20}}{f'_{10}} \frac{f_{10}}{f_{20}}\}} = \frac{-\gamma_o N}{\lambda_m \pi} \frac{f'_{10}/f_{10}}{F_o} \tag{13}$$

Using constants C_{m1}, C_{m2} given by eqn. (12) and (13), we get

$$r \tilde{w}_\gamma = \frac{\partial \tilde{\Phi}_\gamma}{\partial \theta} = -\frac{N}{\pi} \int_o^\xi \gamma_o \sum_{m=1}^{\infty} \frac{1}{F_o} \frac{f'_{10}}{f_{10}} \frac{f_2}{f_{20}} \cos \lambda_m (\theta - \theta_o) \, d\xi_o -$$

$$\frac{N}{\pi} \int_\xi^\ell \gamma_o \sum_{m=1}^{\infty} \frac{1}{F_o} \frac{f'_{20}}{f_{20}} \frac{f_1}{f_{10}} \cos \lambda_m (\theta - \theta_o) \, d\xi_o \tag{14}$$

and $$\frac{\tilde{v}_\gamma}{v_{p*}} = -\frac{N}{\pi} \int_o^\xi \gamma_o \sum_{m=1}^{\infty} \frac{1}{\lambda_m F_o} \frac{f'_{10}}{f_{10}} \frac{f'_2}{f_{20}} \sin \lambda_m (\theta - \theta_o) \, d\xi_o -$$

$$\frac{N}{\pi} \int_\xi^\ell \gamma_o \sum_{m=1}^{\infty} \frac{1}{\lambda_m F_o} \frac{f'_{20}}{f_{20}} \frac{f'_1}{f_{10}} \sin \lambda_m (\theta - \theta_o) \, d\xi_o \tag{15}$$

Referring to fig. 3(a), velocities in ξ and η direction can be written as,

$$\tilde{C}_{\xi} = \tilde{v} \cos \varepsilon + \tilde{w} \sin \varepsilon \tag{16}$$

$$\tilde{C}_{\eta} = \tilde{v} \sin \varepsilon + \tilde{w} \cos \varepsilon \tag{17}$$

With the help of eqns. (16) and (17) and using eqns. (7), (8), (14) and (15), eqns. (2.18) to (2.21) can be obtained directly.

APPENDIX - III

CORRECTION FOR SMALL DEPARTURES FROM "FREE-VORTEX" MEAN FLOW

When an infinite number of blades is present, one can view the flow in two ways :

(i) As being an irrotational, unsteady and non-axisymmetric potential flow with an infinite number of embedded sheets of vorticity shed by the variation of lift along the blade span.

(ii) As a steady, axisymmetric, rotational mean flow (averaged in θ-direction), with the blade forces represented as a body force field. The latter view is taken here.

The momentum equation :

$$\frac{D\vec{V}}{Dt} = -\frac{\nabla p}{\rho} + \vec{F} \qquad (1)$$

Where $\vec{V}$ is the velocity vector, t is the time variable, p is the pressure field, ρ is the density of the fluid and $\vec{F}$ is body force per unit mass of the fluid, is easily transformed to Crocco's equation for steady flow :

$$\vec{\zeta} \times \vec{V} = -\nabla H + \vec{F} \qquad (2)$$

where $\vec{\zeta}$ is the vorticity vector and is given by,

$$\vec{\zeta} = \nabla \times \vec{V}$$

and

$$H = \frac{p}{\rho} + \frac{V^2}{2}$$

where H is the total enthalpy for an isentropic incompressible flow.

In (α,β,θ) coordinate frame, the α-component of this eqn. (2) is,

$$(\vec{\zeta} \times \vec{V})_\alpha = - \frac{\partial H}{\partial S_\alpha} \tag{3}$$

Since F_α, the blade force in α-direction has been assumed to be negligible, the eqn.(3) can be written as,

$$\zeta_\beta \; w_* - \zeta_\theta \; v_* = - \; r \; v_{p*} \frac{\partial H}{\partial \alpha} \tag{4}$$

as $$dS_\alpha = \frac{d\alpha}{r \; v_{p*}}$$

Also in the curvilinear coordinate system (α,β,θ), we have

$$\zeta_\theta = (\nabla \times \vec{V})_\theta = r \; v_{p*}^2 \left[\frac{\partial}{\partial \alpha} \left(\frac{v_*}{v_{p*}} \right) - \frac{\partial}{\partial \beta} \left(\frac{u_*}{r \; v_{p*}} \right) \right] \tag{5}$$

$$\zeta_\beta = (\nabla \times \vec{V})_\beta = v_{p*} \left[\frac{\partial}{\partial \theta} \left(\frac{u_*}{r \; v_{p*}} \right) - \frac{\partial}{\partial \alpha} (r w_*) \right]$$

$$= - \; v_{p*} \frac{\partial}{\partial \alpha} (r w_*) \tag{6}$$

as the first term is zero for axisymmetric flow.

Using eqns. (5) and (6), eqn. (4) can be written as follows :

$$w_* \; v_{p*} \frac{\partial}{\partial \alpha} (r w_*) + v_* \; r \; v_{p*}^2 \left[\frac{\partial}{\partial \alpha} \left(\frac{v_*}{v_{p*}} \right) - \frac{\partial}{\partial \beta} \left(\frac{u_*}{r \; v_{p*}} \right) \right] =$$

$$r \; v_{p*} \frac{\partial H}{\partial \alpha} \tag{7}$$

From the Euler's turbine equation, we have

$$dH = \omega \, d \, (r w_*) \tag{8}$$

Assuming that the flow is a "simple equilibrium flow" i.e. a flow approximately along the α = constant surfaces, u_* can be neglected as compared to v_* or v_{p*}. So the eqn.(7) reduces to,

$$\frac{1}{2}\frac{\partial}{\partial\alpha}\left(\frac{v_*}{v_{p*}}\right)^2 = \frac{1}{\frac{2}{v_{p*}}}\left[\omega - \frac{w_*}{r}\right]\frac{\partial}{\partial\alpha}(r\,w_*) \qquad (9)$$

Substituting $\kappa = (v_* - v_{p*})/v_{p*}$ into eqn.(9), and neglecting the terms of the order of κ^2, as it is assumed that the change in v_* from v_{p*} will be small, we get

$$\frac{\partial}{\partial\alpha}(\kappa) = \frac{1}{\frac{2}{v_{p*}}}\left[\omega - \frac{w_*}{r}\right]\frac{\partial}{\partial\alpha}(r\,w_*)$$

or
$$\kappa = \int_{\alpha_h}^{\alpha} \frac{1}{\frac{2}{v_{p*}}}\left[\omega - \frac{w_*}{r}\right]\frac{\partial}{\partial\alpha}(r\,w_*)\,d\alpha + \kappa_h \qquad (10)$$

where the subscript h denotes the value at the hub. The volume flow Q through the passage of turbomachine can be expressed as follows :

$$Q = \int_{\text{hub}}^{\text{tip}} 2\pi\, r\, v_*\, d\, S_\alpha \qquad (11)$$

Now using eqn.(10) and the definition of κ , we get

$$\kappa_h = -\frac{2\pi}{Q}\int_{\alpha_h}^{\alpha_t}\int_{\alpha_h}^{\alpha}\frac{1}{\frac{2}{v_{p*}}}\left[\omega - \frac{w_*}{r}\right]\frac{\partial}{\partial\alpha}(r\,w_*)\,d\alpha\,d\alpha \qquad (12)$$

where the subscript t denotes the value at the casing and we take

$(\alpha_h - \alpha_t) = \frac{Q}{2\pi}$.

APPENDIX IV

THE COMPUTER PROGRAMS

Two computer programs were used for the numerical solution of the inverse problem. They are, (i) For meridional potential duct flow solution, and (ii) For the inverse problem, in which the meridional potential duct flow solution is used as an input.

The computer programs were written in Fortran IV and run on an IBM 7044 computer with approximately 32K words of core storage. The programs are quite general except that some sections will require changes, which will depend on the turbomachine geometry.

Fortran listing of these programs is given in Appendix VI.

1. Description of Program for Meridional Potential Duct Flow Solution :

MAIN Program reads and prints various input and output data, solves the system of simultaneous linear algebraic equations resulting from the finite difference approximation of partial differential eqn. (2.1) and calls various subroutines.

Subroutine Name	Description
BNDPTS	Called by MAIN, locates the boundary points, their associated intercepts, and coefficients C,D,E,F,G appearing in eqn. (1) of Appendix I.
STREAM	Called by MAIN, for determining positions of a given number of streamlines, slope and velocity along them.
DERIV	Called by STREAM, for calculating derivative.

Subroutine Name	Description
SMOOTH	Called by STREAM, for smoothing coordinates of streamlines, slope and velocity along them.

LIST OF PRINCIPAL VARIABLES

FORTRAN Implementation :

Program Symbol	Definition
(MAIN)	
ALPHA, BETA	Grid size of the flow region to be analysed.
DX	Mesh size, Δx.
D,E,F,G	Coefficients D,E,F,G defined by eqn. (1) in Appendix I, and multiplied by C.
EPS	Sum of percentage variations in the functional values at grid points from iteration to iteration.
EPSMAX	Convergence criteria for EPS.
FJSHFT, FISHFT	Shifts in coordinate axes.
IR	Matrix such that IR(K,J) contains the extreme column subscript of a completely interior point in row J near boundary K.
IMAX	Matrix such that IMAX (K,J) contains the column subscripts I of the first and last boundary points (of types I-IV) in row J near boundary K.
ITYPE	Matrix giving type of each boundary point (type I to IV), shown in fig. 1(b).
ITER	Counter on the number of Gauss-Seidal iteration.
ITMAX	Maximum allowable value of ITER.

Program Symbol	Definition
I,J	Grid-point subscripts i,j.
K	Subscript in IMAX and IR for inner and outer boundaries.
M,N	Number of intervals, in which ALPHA and BETA are divided.
NS	Number of stream surfaces.
PSI	Matrix containing values of ψ at each grid point.
PSIMAX	Maximum value of ψ i.e. at the casing.
RF	Relaxation factor.
(Subroutine BNDPTS)	
A,B	Intercepts near the boundary grid point in BETA and ALPHA directions.
EPSML	Very small distance.
(Subroutine STREAM)	
A	Vector containing values of ψ at given number of stream surfaces.
AA,BB	Vectors containing function and independent variable for calculating derivatives.
AP, BP	Values of derivatives.
NII	Vector containing the value of number of points on each stream surface.
NI,NO	Counters on the number of points on a stream surface.
SLOPE	Slope of the meridional streamline.
TSLP	True slope of known streamlines.

Program Symbol	Definition
V	Meridional velocity along a streamline.
VR, VZ	Velocity components of V.
X, Y	Matrix containing coordinates of streamlines.
(Subroutine SMOOTH)	
VI, XI, YI, SI	Smoothed values of V, X, Y and SLOPE at a point.
TEMP	Temporary storage for old values used for smoothing.

2. Description of Program for the Inverse Problem :

MAIN Program reads and prints various input data, some output data and calls various subroutines.

Subroutine Name	Description
STRIM	Called by MAIN, calculates lengths of various streamlines from the inlet edge and non-dimensionalises various input data.
SSS	Function used for calculating length of a curve between any two points.
VGIVEN	Called by MAIN, interpolates values of specified velocity distribution from the input data for the required number of points.
SHAPE	Called by MAIN, calculates initial shape of the camber lines.
RADIUS	Function used for selection and/or interpolation for a value of r corresponding to any value of S_β.

Subroutine Name	Description
VPPP	Function used for selection and/or interpolation for a value of v_{p*} corresponding to any value of r.
AREA	Function used for integration between any two points.
WKBAPR	Called by MAIN, calculates various variables appearing in WKB solution.
DERIV, PRIME	Called by WKBAPR, used for calculation of derivatives.
YKASHI	Called by MAIN, calculates induced velocities due to vortices.
SRDTBN	Called by MAIN, calculates the source distribution on stream surfaces.
VSOURC	Called by MAIN, calculates induced velocities due to sources and sinks.
BLADE	Called by MAIN, calculates the camber line shape, Blade thickness, estimates v_* from v_{p*} and checks the convergence of blade profile.

LIST OF PRINCIPAL VARIABLES

FORTRAN Implementation Program Symbol (MAIN)	Definition
A1, A2, A3, AI	Vectors used for different temporary storages.
APSI	Vector contain values of ψ at a number of stream surfaces.
BTH	Blade thickness.

Program Symbol	Definition
CLL	Matrix containing lengths along the camber lines ξ.
CGI	Specified velocity distribution.
CGIGMC, CETGMC	Induced velocities due to vortices in ξ and η direction.
CGIQC, CETQC	Induced velocities due to sources and sinks in ξ and η directions.
D	Constants B_n from eqn. (4.11).
DMY	Dummy variable.
DTDGI	$dt/d\xi$.
EPS	Angle ε.
F1, F2, F1P, F2P, FO	Functions defined by eqns. (5.8), (5.9) and (5.15).
GYL	ξ/ℓ.
ITER	Counter used for number of iterations.
LMDA	λ_m.
M	Maximum number of terms can be taken for infinite series summation used for calculating induced velocities due to singularities.
NBLADE	Number of blades.
NS	Number of stream surfaces.
NF	Number of terms used in Fourier's Approximation.
NII	Number of points known at each streamline.
NUMBER	Number of points at which specified velocity distribution at each stream surface is given as an input.

Program Symbol	Definition
P, PP	Functions A and A' in eqn. (5.9).
PSIMAX	Maximum value of ψ, i.e. at the casing.
RC	Coordinate r of the camber line.
SOURCE	Source strengths of sources placed on camber line, σ.
SCBETA	Length S_β for different points on the camber line.
SBETA	Length of streamline at different points from the leading edge.
THETA	Coordinate θ.
TAU	Function τ.
VP	v_{p*}.
VSTAR	v_*.
VORTEX	Strength of vortices placed on the camber line, γ.
V	Meridional potential flow velocity.
X	Z-coordinate of streamlines.
XX	Value of ξ/ℓ for the points at which velocity distribution is specified as an input.
Y	Coordinate r of streamlines.
(Subroutine STRIM)	
RTIP	r_{tip}.
RTIPS	$(r_{tip})^2$.
(Subroutine SHAPE)	
GAMA	Circulation, Γ defined by eqn. (2.42).
R	Coordinate r of streamlines.
TANEPS	Slope of the camber line.

Program Symbol	Definition
(Subroutine WKBAIR)	
A1	Function defined by eqn. (2.12).
G1,G2	Functions defined by eqn. (2.11).
K	Function defined by eqn.(2.14).
F,FP,FPP	Function defined by eqn.(2.10) and its first and second derivatives.
(Subroutine DERIV)	
YP	Derivative.
(Subroutine YKASHI)	
AIGMC	$I_{\gamma c}$, defined by eqn.(2.20).
AJGMC	$J_{\gamma c}$, defined by eqn. (2.21).
BN	I_n, defined by eqn.(5.6).
CON	Convergence criteria for infinite series summation.
FFF	$(f'_{10}f'_{20})/(f_{10}f_{20}F_o)$, defined by eqns. (2.20) or (2.21).
FOYFP	f_o/f', defined by eqn. (5.14).
FPOYFP	f'_o/f', defined by eqn. (5.12).
VAL	Absolute value of last term of the series to be added in the series summation.
(Subroutine SRDTBN)	
AIQC	I_{qc}, defined by eqn. (2.18).
AJQC	J_{qc}, defined by eqn. (2.19).
AJQCYX	$J_{qc}/(x-x_o)$.

Program Symbol	Definition
BN	I_n, defined by eqn.(5.6).
CGI/CCG	Induced velocity due to cascade of sources, defined by eqn. (2.35).
ES	E, defined by eqn. (2.37).
ER	Percentage change in value of source strength at some point.
ERROR	Cumulative percentage change in value of source strength at all points on a camber line.
FOYF	f_o/f, defined by eqn.(5.13).
FPOYF	f'_o/f, defined by eqn.(5.10).
ITER	Counter for iteration in the solution of integral eqn. (2.40).
T	Function defined by eqn. (2.26).

(Subroutine VSOURC)

Symbols used are same as already defined above.

(Subroutine BLADE)

AK	κ, defined by eqn. (2.49).
ALPHA	α, defined by eqn.(2.52).
ERROR	Cumulative percentage variation at all points on a camber line.
FK	Function inside the integral in eqn. (2.49).
FKH	κ_h, defined by eqn. (2.51).
GAMA	Circulation, Γ defined by eqn.(2.42).
RW	Product $(r\, w_x)$.
SAVBTH	Blade thickness of the previous iteration.
STHETA	θ-coordinate of previous iteration.

APPENDIX V

INPUT DATA FOR THE NUMERICAL EXAMPLE

Various input data used for the numerical example given here, are given below. Names of these variables are same as used in Fortran program.

Meridional Potential duct flow Solution :

A	Stream function values at various stream surfaces are: 0.0(I), 0.5345(II), 1.069(III), 1.6035(IV), 2.138(V).
ALPHA	80 mm.
BETA	100 mm.
EPSMAX	0.01
ITMAX	400
M	40
N	50
NS	5
RF	R_f, 1.8

Inverse Problem

APSI	Same as A given above.
CLL	Total camber line lengths in non-dimensional form are : 1.75(I), 1.33(II), 1.05(III), 0.855(IV), 0.65(V).
M	Maximum number of terms for infinite series summation, 11
NBLADE	N, 8
NF	17
NUMBER	13
PSIMAX	2.138 (non-dimensional)
RTIP	r_{tip}, 67 mm.

APPENDIX VI

```
C*************************************************************************
C***** AXISYMMETRIC POTENTIAL FLOW ANALYSIS FOR A MIXED FLOW TURBOMACHINE.
C.........................................................................
C..... DESCRIPTION OF THE PROGRAM ARE GIVEN IN APPENDIX I AND IV.
C..... MAIN PROGRAM.
C.........................................................................
C*************************************************************************
C
      DIMENSION IR(2,51),IMAX(2,51),ITYPE(26,36),D(26,36),E(26,36),
     1F(26,36),G(26,36),PSI(41,51)
      READ 100,M,N,ITMAX,ALPHA,BETA,EPSMAX,RF
      PRINT 101,M,N,ITMAX,ALPHA,BETA,EPSMAX,RF
      MP1=M+1
      NP1=N+1
      FN=N
      DX=BETA/FN
      READ 105,NS
      PSIMAX=2.138
      FJSHFT=25.0
      FISHFT=13.0
      DO 1 I=1,MP1
      DO 1 J=1,NP1
    1 PSI(I,J)=0.5*PSIMAX
      DO 2 I=25,41
    2 PSI(I,1)=0.0
      DO 3 J=36,51
    3 PSI(13,J)=PSIMAX
      DO 4 J=25,51
    4 PSI(1,J)=0.0
      DO 5 I=26,41
    5 PSI(I,23)=PSIMAX
      APS=PSIMAX/12.
      DO 6 I=2,12
    6 PSI(I,51)=PSI(I-1,51)+APS
      AX=67.**2-23.**2
      DO 7 J=2,22
      FJM1=J-1
      R2=FJM1*DX+FJSHFT
      R1=R2-DX
      DPSI=(R2*R2-R1*R1)*PSIMAX/AX
      PSI(41,J)=PSI(41,J-1)+DPSI
    7 CONTINUE
      IMAX(1,1)=25
      IMAX(2,1)=41
      IMAX(1,51)=1
      IMAX(2,51)=13
      CALL BNDPTS(M,N,MP1,NP1,IR,IMAX,ITYPE,D,E,F,G,DX,FJSHFT,FISHFT)
      PRINT 101
      PRINT 106,(IR(1,J),J=2,N)
      PRINT 102
      PRINT 107,(IMAX(1,J),J=1,NP1)
      PRINT 110
```

```
      WRITE(6,111)
      WRITE(6,112,(IMAX(2,J),J=1,NP1)
      IT=0
    9 IT=IT+1
      EPS=0.0

C
C     INTERIOR POINTS
C
      DO 8 J=2,J
      IHIGH1=IR(1,J)
      IHIGH2=IR(2,J)
      DO 8 I=IHIGH1,IHIGH2
      SAVPSI=PSI(I,J)
      FJ1=J-1
      FJ=FJ1+FJSHFT/DX
      PSI(I,J)=(PSI(I-1,J)+PSI(I+1,J)+PSI(I,J-1)+PSI(I,J+1))*0.25-0.5*(
     1PSI(I,J+1)-PSI(I,J-1))/(4.*FJ)
      PSI(I,J)=SAVPSI+RF *(PSI(I,J)-SAVPSI)
      IF(ABS(SAVPSI).LT..1E-10)GO TO 8
      EPS=EPS+ABS((PSI(I,J)-SAVPSI)/SAVPSI)*100.
    8 CONTINUE

C
C     BOUNDRY POINTS
C
      DO 14 K=1,2
      IF(K.EQ.1)GO TO 14
C.....BOUNDRY POINTS ON CASING(OUTER BOUNDRY),BOUNDRY CURVED CONVEX
C     TOWARDS ORIGIN OF COORDINATE SYSTEM.
      DO 13 J=23,35
      IHIGH2=IMAX(K,J)
      ILOW2=IR(K,J)+1
      DO 13 I=ILOW2,IHIGH2
      IF(ITYPE(I,J).EQ.4)GO TO 13
      SUM=D(I,J)*PSI(I-1,J)+E(I,J)*PSI(I,J-1)
      IF(ITYPE(I,J).NE.2)GO TO 10
      SUM=SUM+G(I,J)*PSI(I+1,J)+PSIMAX*F(I,J)
      GO TO 12
   10 IF(ITYPE(I,J).NE.3)GO TO 11
      SUM=SUM+F(I,J)*PSI(I,J+1)+PSIMAX*G(I,J)
      GO TO 12
   11 SUM=SUM+PSIMAX*(G(I,J)+F(I,J))
   12 SAVPSI=PSI(I,J)
      PSI(I,J)=SUM
      PSI(I,J)=SAVPSI+RF *(PSI(I,J)-SAVPSI)
      IF(ABS(SAVPSI).LT..1E-10)GO TO 13
      EPS=EPS+ABS((PSI(I,J)-SAVPSI)/SAVPSI)*100.
   13 CONTINUE
      GO TO 16
C.....BOUNDRY POINTS ON HUB(INNER BOUNDRY),BOUNDRY CURVED CONVEX TOWARDS
C     ORIGIN OF COORDINATE SYSTEM.
   14 DO 17 J=2,24
      IHIGH1=IMAX(K,J)
      ILOW1=IR(K,J)-1
      DO 17 I=IHIGH1,ILOW1
```

```
      DIMENSION IR(2,51),IMAX(2,51),ITYPE(26,36),D(26,36),E(26,36),
     1F(26,36),G(26,36)
C
C.....LOCATE EXTREME POINTS
C.....(J,IMAX(K,J)) AND DETERMINATION OF ITS TYPE.
C
      EPSIL=0.1E-05
      IMAX(2,23)=25
      DO 1 J=2,22
      IMAX(2,J)=41
    1 IR(2,J)=40
      DO 2 J=36,51
      IMAX(2,J)=13
    2 IR(2,J)=12
      DO 3 J=25,51
      IMAX(1,J)=1
    3 IR(1,J)=2
      DO 20 K=1,2
      IF(K.EQ.2)GO TO 14
C.....MOVE (INNER BOUNDRY).
      DO 10 J=2,24
      FJM1=J-1
      XOVRDX=(47.-SQRT(47.**2-(47.-FJM1*DX)**2))/DX
      IM1=XOVRDX+EPSML+1.
      FIM1=IM1
      I=IM1+1
      IMAX(K,J)=I
      B=FIM1-XOVRDX
      IF(I.GT.24)GO TO 4
      YOVRDX=(47.-SQRT(47.**2-(47.-FIM1*DX)**2))/DX
      A=FJM1-YOVRDX
      GO TO 5
    4 A=1.0
    5 FJ=FJM1+FJSHFT/DX
      IF(ABS(A).LT.EPSML)GO TO 6
      IF(ABS(B).GE.EPSML)GO TO 7
    6 ITYPE(I,J)=4
      GO TO 10
    7 IF(A.LE.1.0)GO TO 8
      ITYPE(I,J)=3
      A=1.0
      F(I,J)=1./(A*A+A)
      G(I,J)=(1.-1./(2.*FJ))/(B*(B+1.0))
      GO TO 9
    8 ITYPE(I,J)=1
      F(I,J)=1./(A*A+A)
      G(I,J)=(1.-1./(2.*FJ))/(B*(B+1.0))
    9 CONTINUE
      C=1./((A+B)/(A*B)-(1.-B)/(2.*FJ*B))
      D(I,J)=1./(1.+B)+B/(2.*FJ*(1.+B))
      E(I,J)=1./(1.+A)
      D(I,J)=D(I,J)*C
      E(I,J)=E(I,J)*C
      F(I,J)=F(I,J)*C
```

```
 10     GO TO 20
C*****LOCATE BOUNDRY POINTS OF TYPE 2 ON HUB (INNER BOUNDRY).
       DO 13 J=2,24
       FJM1=J-1
       I=IMAX(K,J)+1
 11    IF(IMAX(K,J-1).LE.I)GO TO 12
       FIM1=I-1
       YOVRDX=(47.-SQRT(47.**2-(47.-FIM1*DX)**2))/DX
       A=FJM1-YOVRDX
       B=1.0
       FJ=FJM1+FJSHFT/DX
       ITYPE(I,J)=2
       C=1./((A+B)/(A*B)-(1.-B)/(2.*FJ*B))
       D(I,J)=1./(1.+B)+B/(2.*FJ*(1.+B))
       E(I,J)=1./(1.+A)
       F(I,J)=1./(A*A+A)
       G(I,J)=(1.-1./(2.*FJ))/(B*(B+1.0))
       D(I,J)=D(I,J)*C
       E(I,J)=E(I,J)*C
       F(I,J)=F(I,J)*C
       G(I,J)=G(I,J)*C
       I=I+1
       GO TO 11
 12    IB(K,J)=I
 13    CONTINUE
       GO TO 23
C.....CASING(OUTER BOUNDRY).
 14    DO 22 J=23,35
       FJM1=J-1
       XOVRDX=(49.-SQRT(25.**2-(69.-FJM1*DX)**2))/DX
       IM1=XOVRDX+EPSML
       FIM1=IM1
       I=IM1+1
       IMAX(K,J)=I
       B=XOVRDX-FIM1
       YOVRDX=(69.-SQRT(25.**2-(49.-FIM1*DX)**2))/DX
       A=YOVRDX-FJM1
 15    FJ=FJM1+FJSHFT/DX
       IF(ABS(A).LT.EPSML)GO TO 16
       IF(ABS(B).GE.EPSML)GO TO 17
 16    ITYPE(I,J)=4
       GO TO 20
 17    IF(A.LE.1.0)GO TO 18
       ITYPE(I,J)=3
       A=1.0
       F(I,J)=1./(A*A+A)
       G(I,J)=(1.-1./(2.*FJ))/(B*(B+1.0))
       GO TO 19
 18    ITYPE(I,J)=1
       F(I,J)=1./(A*A+A)
       G(I,J)=(1.-1./(2.*FJ))/(B*(B+1.0))
 19    C=1./((A+B)/(A*B)-(1.-B)/(2.*FJ*B))
       D(I,J)=1./(1.+B)+B/(2.*FJ*(1.+B))
       E(I,J)=1./(1.+A)
```

```
      (I,J)=D(I,J)*C
      E(I,J)=E(I,J)*C
      F(I,J)=F(I,J)*C
      G(I,J)=G(I,J)*C
   23 CONTINUE
C
C.....LOCATE BOUNDRY POINTS OF TYPE 2 ON CASING(OUTER BOUNDRY).
C
      DO 25 J=23,35
      FJM1=J-1
      I=IMAX(K,J)-1
   21 IF(IMAX(K,J+1).GE.I)GO TO 22
      FIM1=I-1
      YOVDX=(49.-SQRT(25.**2-(49.-FIM1*DX)**2))/DX
      A=YOVDX-FJM1
      B=1.
      FJ=FJM1+FJSHFT/DX
      ITYPE(I,J)=2
      C=1./((A+B)/(A*B)-(1.-B)/(2.*FJ*B))
      D(I,J)=1./(1.+B)+B/(2.*FJ*(1.+B))
      E(I,J)=1./(1.+A)
      F(I,J)=1./(A*A+A)
      G(I,J)=(1.-1./(2.*FJ))/(B*(B+1.0))
      D(I,J)=D(I,J)*C
      E(I,J)=E(I,J)*C
      F(I,J)=F(I,J)*C
      G(I,J)=G(I,J)*C
      I=I-1
      GO TO 21
   22 IR(K,J)=I
   23 CONTINUE
      DO 27 J=2,35
      L1=IMAX(1,J)
      L2=IR(1,J)-1
      DO 25 I=L1,L2
      IF(ITYPE(I,J).EQ.4)GO TO 24
      PRINT 100,J,I,ITYPE(I,J),D(I,J),E(I,J),F(I,J),G(I,J)
      GO TO 25
   24 PRINT 100,J,I,ITYPE(I,J)
   25 CONTINUE
      M2=IR(2,J)+1
      M1=IMAX(2,J)
      IF(M1.GE.26)GO TO 27
      DO 27 I=M2,M1
      IF(ITYPE(I,J).EQ.4)GO TO 26
      PRINT 100,J,I,ITYPE(I,J),D(I,J),E(I,J),F(I,J),G(I,J)
      GO TO 27
   26 PRINT 100,J,I,ITYPE(I,J)
   27 CONTINUE
  100 FORMAT(2X,3I5,4E20.8)
      RETURN
      END
C*************************************************************
C*****SUBROUTINE FOR DETERMINING POSITIONS OF A GIVEN NUMBER OF
C*****STREAMLINES,VELOCITY AND SLOPE.
C*************************************************************
```

```
      SUBROUTINE STREAM(PSI,IMAX,PSIMAX,DX,NS,FJSHFT,FISHFT)
      DIMENSION PSI(41,51),IMAX(2,51),A(5 ),NII(5 ),X(5 ,80),Y(5 ,80),
     1 V(5 ,80),SLOPE(5 ,80),AA(3),BB(3)
C
      DO 1 I=1,NS
      DO 1 J=1,80
      V(I,J)=0.0
      SLOPE(I,J)=0.0
      X(I,J)=0.0
    1 Y(I,J)=0.0
      DO 2 I=1,NS
    2 NII(I)=0
C.....PSI VALUES SPECIFIED ON EACH STREAM SURFACE.
      A(1)=0.0
      A(2)=PSIMAX/4.
      A(3)=2.*A(2)
      A(4)=3.*A(2)
      A(5)=PSIMAX
      PRINT 101
      PRINT 103,(A(N),N=1,NS)
      DO 3 L=1,27
      J=50-L
      FJ=J-1
      FJ=FJ*DX+FJSHFT
      AA(1)=0.0
      AA(2)=DX
      AA(3)=2.0*DX
      BB(1)=0.0
      BB(2)=PSI(2,J)
      BB(3)=PSI(3,J)
      CALL DERIV(AA,BB,BP)
      VR=(1./FJ)*BP
      V(1,L)=VR
      SLOPE(1,L)=.1E05
      X(1,L)=0.0
    3 Y(1,L)=1.52.-FLOAT(L)*DX
      X(1,28)=1.0
      Y(1,28)=47.0
      SI=0.5*(PSI(2,25)+PSI(2,24))
      BB(2)=SI
      BB(3)=0.5*(PSI(3,25)+PSI(3,24))
      CALL DERIV(AA,BB,BP)
      V(1,28)=(1./70.)*BP
      SLOPE(1,28)=.1E05
      NI=28
      DXXX=DX
      DXXXX=DX
      DO 10 L=2,24
      J=26-L
      NI=NI+1
      FJ=J-1
      Y(1,NI)=FJ*DX
      X(1,NI)=47.-SQRT(47.**2-(47.0-FJ*DX)**2)
      XX=X(1,NI)/DX+1.0
```

```
      J=J+NJ+FJSHFT
      FJ=J-1
      IF(ABS(DXX).LT..1E-1)GO TO 4
      AA(2)=DX
      AA(3)=DX+DXX
      BB(2)=PSI(IX,J)
      BB(3)=PSI(IX+1,J)
      CALL DERIV(AA,BB,BP)
      VR=(1./FJ)*BP
      GO TO 5
4     AA(2)=DX
      AA(3)=2.0*DX
      BB(2)=PSI(IX+1,J)
      BB(3)=PSI(IX+2,J)
      CALL DERIV(AA,BB,BP)
      VR=(1./FJ)*BP
      BB(2)=PSI(IX,J+1)
      BB(3)=PSI(IX,J+2)
      CALL DERIV(AA,BB,AP)
      VZ=-(1./FJ)*AP
      GO TO 7
5     IF(IMAX(I,J+1).GE.IMAX(I,J))GO TO 6
      IX1=IX-1
      SI=PSI(IX,J+1)-(PSI(IX,J+1)-PSI(IX1 ,J+1))*DXX/DX
      AA(2)=DX
      AA(3)=2.0*DX
      BB(2)=SI
      BB(3)=PSI(IX,J+2)-(PSI(IX,J+2)-PSI(IX1,J+2))*DXX/DX
      CALL DERIV(AA,BB,AP)
      VZ=-(1./FJ)*AP
      GO TO 7
6     D=DXX/DXXX
      AA(2)=DX
      AA(3)=2.0*DX
      BB(2)=PSI(IX,J+1)*(1.-D)
      IF(IMAX(I,J+2).GE.IMAX(I,J))GO TO 61
      DD=DXX/DX
      J2=J+2
      BB(3)=PSI(IX-1,J2)+(PSI(IX,J2)-PSI(IX-1,J2))*(1.-DD)
      CALL DERIV(AA,BB,AP)
      VZ=-(1./FJ)*AP
      GO TO 7
61    DD=DXX/DXXXX
      BB(3)=PSI(IX,J+2)*(1.-DD)
      CALL DERIV(AA,BB,AP)
      VZ=-(1./FJ)*AP
7     V(I,MI)=SQRT(VR**2+VZ**2)
      IF(ABS(VZ).LT..1E-05)GO TO 8
      SLOPE(I,MI)=VR/VZ
      GO TO 9
8     SLOPE(I,MI)=.1E05
9     DXXXX=DXXX
      DXXX=DXX
      DXX=DX
      RETURN
      END
```

```
      X(1,NI)=47.0
      Y(1,NI)=69.0
      SI=0.5*(PSI(25,2)+PSI(24,2))
      AA(2)=DX
      AA(3)=2.0*DX
      BB(2)=SI
      BB(3)=0.5*(PSI(25,3)+PSI(24,3))
      CALL DERIV(AA,BB,AP)
      V(1,NI)=(1./23.)*AP
      SLOPE(1,NI)=0.0
      DO 11 L=1,17
      NI=NI+1
      I=24+L
      BB(2)=PSI(I,2)
      BB(3)=PSI(I,3)
      CALL DERIV(AA,BB,AP)
      V(1,NI)=(1./23.)*AP
      SLOPE(1,NI)=0.0
      X(1,NI)=46.+FLOAT(L)*DX
11    Y(1,NI)=69.
      NII(1)=NI
      DO 12 L=1,16
      J=52-L
      FJ=J-1
      FJ=FJ*DX+FJSHFT
      AA(1)=2.0*DX
      AA(2)=DX
      AA(3)=0.0
      BB(1)=PSIMAX
      BB(2)=PSI(12,J)
      BB(3)=PSI(11,J)
      CALL DERIV(AA,BB,BP)
      V(NS,L)=(1./FJ)*BP
      SLOPE(5,L)=.1E05
      X(5,L)=24.
12    Y(5,L)=102.-FLOAT(L)*DX
      X(5,17)=24.
      Y(5,17)=69.
      SI=(PSI(12,36)+PSI(12,35))*0.5
      BB(2)=SI
      BB(3)=0.5*(PSI(11,36)+PSI(11,35))
      CALL DERIV(AA,BB,BP)
      V(NS,17)=(1./92.)*BP
      SLOPE(5,17)=.1E05
      NC=17
      DO 20 L=1,12
      J=36-L
      NC=NC+1
      FJ=J-1
      FJN1=J-1
      X(5,NC)=49.0-SQRT(25.**2-(69.0-FJN1*DX)**2)
      Y(5,NC)=FJ*DX
      IF(J.LE.25)GO TO 13
```

```
      LX2=X2/DX
      DXXX=X2-FLOAT(LX2)*DX
      FJN3=J-3
      X3=69.-SQRT(25.**2-(69.-FJN3*DX)**2)
      LX3=X3/DX
      DXXXX=X3-FLOAT(LX3)*DX
      GO TO 14
 13   IF(J.LE.24)GO TO 131
      DXXX=DXXXX
      DXXX=DX
      GO TO 14
 131  DXXX=DX
      DXXXX=DX
 14   LX=X(5,80)/DX
      DXX=X(5,80)-FLOAT(LX)*DX
      FJ=FJ*DX+FJSHFT
      IX=LX+1
      IF(ABS(DXX).LT..1E-1)GO TO 15
      AA(1)=DX+DXX
      AA(2)=DX
      BB(2)=PSI(IX,J)
      BB(3)=PSI(IX-1,J)
      CALL DERIV(AA,BB,BP)
      VR=(1./FJ)*BP
      GO TO 16
 15   AA(1)=2.0*DX
      AA(2)=DX
      BB(2)=PSI(IX-1,J)
      BB(3)=PSI(IX-2,J)
      CALL DERIV(AA,BB,BP)
      VR=(1./FJ)*BP
      BB(2)=PSI(IX,J-1)
      BB(3)=PSI(IX,J-2)
      CALL DERIV(AA,BB,AP)
      VZ=-(1./FJ)*AP
      GO TO 18
 16   IF(IMAX(2,J-1).LE.IMAX(2,J))GO TO 17
      IX1=IX+1
      SI=PSI(IX,J-1)+(PSI(IX1 ,J-1)-PSI(IX,J-1))*DXX/DX
      AA(1)=2.0*DX
      AA(2)=DX
      BB(2)=SI
      BB(3)=PSI(IX,J-2)+(PSI(IX1,J-2)-PSI(IX,J-2))*DXX/DX
      CALL DERIV(AA,BB,AP)
      VZ=-(1./FJ)*AP
      GO TO 18
 17   D=DXX/DXXX
      AA(1)=2.0*DX
      AA(2)=DX
      BB(2)=PSI(IX,J-1)+(PSIMAX-PSI(IX,J-1))*D
      IF(IMAX(2,J-2).LE.IMAX(2,J))GO TO 171
      DD=DXX/DX
      J2=J-2
      BB(3)=PSI(IX,J2)+(PSI(IX+1,J2)-PSI(IX,J2))*DD
```

```
      VZ=-(1./FJ)*AP
      GO TO 18
  171 DD=DXX/DXXXX
      J2=J-2
      BB(3)=PSI(IX,J2)+(PSIMAX-PSI(IX,J2))*DD
      CALL DERIV(AA,BB,AP)
      VZ=-(1./FJ)*AP
   1  V(5,NO)=SQRT(VR**2+VZ**2)
      IF(ABS(VZ).LT..1E-05)GO TO 19
      SLOPE(5,NO)=VR/VZ
      GO TO 20
   19 SLOPE(5,NO)=.1E05
   20 CONTINUE
      NC=NO+1
      SI=0.5*(PSI(25,22)+PSI(26,22))
      AA(1)=2.*DX
      AA(2)=DX
      BB(2)=SI
      BB(3)=0.5*(PSI(25,21)+PSI(26,21))
      CALL DERIV(AA,BB,AP)
      V(5,NO)=(1./67.)*AP
      SLOPE(5,NO)=0.0
      X(5,NO)=49.
      Y(5,NO)=44.
      DO 21 L=1,16
      NC=NO+1
      I=25+L
      BB(2)=PSI(I,22)
      BB(3)=PSI(I,21)
      CALL DERIV(AA,BB,AP)
      V(5,NO)=(1./67.)*AP
      SLOPE(5,NO)=0.0
      X(5,NO)=(I-1)*2
   21 Y(5,NO)=44.0
      NII(NS)=40
      NS1=NS-1
      DO 22 I=2,NS1
   22 NII(I)=0
      DO 26 L=2,51
      J=53-L
      INN=IMAX(1,J)
      IOUT=IMAX(2,J)
      IF(J.LE.23)IOUT=25
      IOUT1=IOUT-1
      DO 26 I=INN,IOUT1
      IN=I+1
      I1=I-1
      DO 26 N=2,NS1
      IF(PSI(I,J).GT.A(N))GO TO 26
      IF(PSI(IN,J).LT.A(N))GO TO 26
      IF(PSI(I,J).EQ.A(N))GO TO 23
      D=PSI(IN,J)-PSI(I,J)
      DXX=(A(N)-PSI(I,J))/D
      NII(N)=NII(N)+1
```

```
      X(N,NA)=(FI+DXX)*DX
      Y(N,NA)=(J-1)*2
      C1=0.5*(PSI(IN,J+1)-PSI(IN,J-1))
      C2=0.5*(PSI(I,J+1)-PSI(I,J-1))
      C=C2+DXX*(C1-C2)
      IF(J.EQ.61)C=PSI(I,J)-PSI(I,J-1)+DXX*(PSI(IN,J)-PSI(IN,J-1)-
     1PSI(I,J)+PSI(I,J-1))
      GO TO 24
   23 NII(N)=NII(N)+1
      NA=NII(N)
      X(N,NA)=FI*DX
      Y(N,NA)=FLOAT(J-1)*DX
      D=0.5*(PSI(I+1,J)-PSI(I-1,J))
      C=0.5*(PSI(I,J+1)-PSI(I,J-1))
      IF(J.EQ.66)C=PSI(I,J)-PSI(I,J-1)
   24 FJ=J-1
      FJ=FJ*DX+FJSHFT
      VR=(1./FJ)*(-D/DX)
      VZ=-(1./FJ)*(C/DX)
      V(N,NA)=SQRT(VR**2+VZ**2)
      IF(ABS(VZ).LT..1E-05)GO TO 25
      SLOPE(N,NA)=VR/VZ
      GO TO 26
   25 SLOPE(N,NA)=.1E05
   26 CONTINUE
      DO 30 I=25,41
      DO 30 J=1,22
      JN=J+1
      FJ=J-1
      DO 30 N=2,MS1
      IF(PSI(I,J).GT.A(N))GO TO 30
      IF(PSI(I,JN).LT.A(N))GO TO 30
      IF(PSI(I,J).EQ.A(N))GO TO 27
      C=PSI(I,JN)-PSI(I,J)
      CYY=(A(N)-PSI(I,J))/C
      NII(N)=NII(N)+1
      NA=NII(N)
      X(N,NA)=(I-1)*2
      Y(N,NA)=(FJ+CYY)*DX
      D1=0.5*(PSI(I+1,JN)-PSI(I-1,JN))
      D2=0.5*(PSI(I+1,J)-PSI(I-1,J))
      D=D2+CYY*(D1-D2)
      IF(I.EQ.41)D=PSI(I,J)-PSI(I-1,J)+CYY*(PSI(I,JN)-PSI(I-1,JN)-
     1PSI(I,J)+PSI(I-1,J))
      GO TO 28
   27 NII(N)=NII(N)+1
      NA=NII(N)
      X(N,NA)=(I-1)*2
      Y(N,NA)=FJ*DX
      D=0.5*(PSI(I+1,J)-PSI(I-1,J))
      C=0.5*(PSI(I,J+1)-PSI(I,J-1))
      IF(I.EQ.41)D=PSI(I,J)-PSI(I-1,J)
   28 FJ=FJ*DX+FJSHFT
      VR=(1./FJ)*(-D/DX)
```

```
      V(N,NA)=SQRT(VR**2+VZ**2)
      IF(ABS(VZ).LT..1E-05)GO TO 29
      SLOPE(N,NA)=VR/VZ
      GO TO 30
   29 SLOPE(N,NA)=.1E05
   30 CONTINUE
      DO 31 N=1,NS
   31 PRINT 102,N,NII(N)
      CALL SMOOTH(X,Y,V,SLOPE,NII,NS)
      PRINT 108
      PRINT 107
      DO 32 N=1,NS
      NA=NII(N)
      PRINT 106,N,NA
      DO 32 K=1,NA
      IF(N.NE.1)GO TO 33
      IF(Y(N,K).LT.1..OR.Y(N,K).GT.46.)GO TO 35
      TSLP=-(X(N,K)-47.)/(Y(N,K)-47.)
      DIFF=(TSLP-SLOPE(N,K))*100./TSLP
      PRINT 103, X(N,K),Y(N,K),V(N,K),SLOPE(N,K),TSLP,DIFF
       GO TO 32
   33 IF(N.NE.NS)GO TO 35
      IF(Y(N,K).LT.45..OR.Y(N,K).GT.68.)GO TO 35
      TSLP=-(X(N,K)-49.)/(Y(N,K)-69.)
      DIFF=(TSLP-SLOPE(N,K))*100./TSLP
      PRINT 103, X(N,K),Y(N,K),V(N,K),SLOPE(N,K),TSLP,DIFF
       GO TO 32
   35 PRINT 104, X(N,K),Y(N,K),V(N,K),SLOPE(N,K)
   32 CONTINUE
C
  101 FORMAT(2X,*PSI VALUES ON STREAM SURFACES ARE.....*)
  102 FORMAT(/,2X,*FINAL VALUE OF NII(*,I2,*)=*,I3)
  103 FORMAT(2X,6E20.8)
  104 FORMAT(4E20.8)
  105 FORMAT(16I5)
  106 FORMAT(/,2X,*SET NO.*,I3,3X,*,NO. OF POINTS IN THIS SET=*,I5)
  107 FORMAT(///,7X,*VALUE OF X*,10X,*VALUE OF Y*,7X,*MERIDIONAL VELOCIT
     1Y*,10X,*SLOPE*)
  108 FORMAT(/ 2X,*STREAM LINES AFTER SMOOTHING .....*)
      RETURN
      END
C*************************************************************
C******SUBROUTINE FOR CALCULATION OF DERIVATIVE.
C*************************************************************
      SUBROUTINE DERIV(AA,BB,BP)
      DIMENSION AA(3),BB(3)
      E0=BB(1)/((AA(1)-AA(2))*(AA(1)-AA(3)))
      E1=BB(2)/((AA(2)-AA(1))*(AA(2)-AA(3)))
      E2=BB(3)/((AA(3)-AA(1))*(AA(3)-AA(2)))
      B0=E0+E1+E2
      B1=-(E0*(AA(2)+AA(3))+E1*(AA(1)+AA(3))+E2*(AA(1)+AA(2)))
      BP=2.0*B0*AA(1)+B1
      RETURN
      END
```

```
C*****SUBROUTINE FOR SMOOTHING STREAMLINES,VELOCITY AND SLOPE.
C**********************************************************************
      SUBROUTINE SMOOTH(X,Y,V,SLOPE,NII,NS)
      DIMENSION X(NS,80),Y(NS,80),V(NS,80),SLOPE(NS,80),NII(NS),TEMP(80)
      DO 11 LS=1,2
      DO 10 I=1,NS
      NA=NII(I)
      NB=NA-2
      DO 1 J=3,NB
      VM=-0.00174*V(I,J-2)+0.1875*V(I,J-1)+0.62847*V(I,J)+0.1875*
     1V(I,J+1)-0.00174*V(I,J+2)
1     TEMP(J)=VM
      DO 2 J=3,NB
2     V(I,J)=TEMP(J)
      GO TO(10,3,3,3,10),I
3     DO 4 J=3,NB
      XM=-0.00174*X(I,J-2)+0.1875*X(I,J-1)+0.62847*X(I,J)+0.1875*
     1X(I,J+1)-0.00174*X(I,J+2)
4     TEMP(J)=XM
      DO 5 J=3,NB
5     X(I,J)=TEMP(J)
      DO 6 J=3,NB
      YM=-0.00174*Y(I,J-2)+0.1875*Y(I,J-1)+0.62847*Y(I,J)+0.1875*
     1Y(I,J+1)-0.00174*Y(I,J+2)
6     TEMP(J)=YM
      DO 7 J=3,NB
7     Y(I,J)=TEMP(J)
      DO 8 J=3,NB
      SM=-0.00174*SLOPE(I,J-2)+0.1875*SLOPE(I,J-1)+0.62847*SLOPE(I,J)+
     1 0.1875*SLOPE(I,J+1)-0.00174*SLOPE(I,J+2)
8     TEMP(J)=SM
      DO 9 J=3,NB
9     SLOPE(I,J)=TEMP(J)
10    CONTINUE
11    CONTINUE
      RETURN
      END
    GOOD

ENCOUNTERED ON SYSTEM INPUT FILE.
----------
ICS -

          000000
          000000
          000753
ED        000921
ED        000000
. ERRORS  000000
----------
```

```
C*****************************************************************
C      MIXED FLOW TURBOMACHINE.
C.....SOLUTION OF THE INVERSE PROBLEM FOR AN HYDRAULIC TURBOMACHINE USIN
C.....SINGULARITY METHOD.
C.....DETAILS OF THE PROGRAM ARE GIVEN IN APPENDIX IV.
C.....MAIN PROGRAM.
C*****************************************************************
      DIMENSION D(17,19),NII(5),LMDM(11),
     2 CGIGMC(5,19),CGTGMC(5,19),CGIQC(5,19),SOURCE(5,19),VSTAR(5,19),
     3TA0(5,19),DMY(5,19),A1(19),A2(19),A3(19),APSI(5)
      DIMENSION  CGTQC(5,19),P(5,19),PP(5,19),DTDGI(5,19),
     4,TH(5,19),X(5,70),Y(5,70),V(5,70),SBETA(5,50),XX(13),CGI(5,2,19),
     5GYL(19),VORTEX(5,19),W(5,19),THETA(5,19),VP(5,19),RC(5,19),AI(19),
     6SCBETA(5,19),EPS(5,19),CLL(5,19),F1(5,11,19),F2(5,11,19),FD(5,11,
     7 19),F1P(5,11,19),F2P(5,11,19)
C
  1     FORMAT(5I5)
  2     FORMAT(4E20.8)
  3     FORMAT(I5,4E20.8)
  4     FORMAT(2X,*STREAMLINE NO.*,I4,*,NO.OF POINTS=*,I4)
  5     FORMAT(10F5.2)
  6     FORMAT(8F10.5)
C
      REAL LMDM
      READ 1,NS
      READ 2,(APSI(I),I=1,NS)
      READ 1,(NII(I),I=1,5)
      READ 6,PSIMAX,FISHFT,FJSHFT
      READ 1,NBLADE,M,NUMBER,NF
      DO 11 I=1,5
      NA=NII(I)
      DO 11 J=1,NA
      K=NA-J+1
      READ 2,X(I,K),Y(I,K),V(I,K)
C.....SHIFTING OF AXES TO THEIR ORIGINAL POSITION.
      X(I,K)=X(I,K)+FISHFT
 11   Y(I,K)=Y(I,K)+FJSHFT
      CALL STRLN(X,Y,V,SBETA,NII)
      DO 12 I=1,5
      NA=NII(I)
      PRINT 4,I,NA
      DO 12 J=1,NA
 12   PRINT 3,J,SBETA(I,J),X(I,J),Y(I,J),V(I,J)
      READ 6,(XX(I),I=1,NUMBER)
      READ 6,(GYL(I),I=1,19)
      DO 13 I=1,5
      DO 13 J=1,2
 13   READ 6,(CGI(I,J,K),K=1,NUMBER)
      CALL VGIVEN(CGI,GYL,XX,A1,NUMBER)
      CALL SHAPE(SBETA,CGI,GYL,Y,V,VORTEX,W,THETA,VP,RC,SCBETA,NBLADE,
     1 EPS,CLL)
      DO 14 I=1,5
      DO 14 J=1,19
```

```
      DMY(I,J)=0.0
      BTH(I,J)=0.0
      DTDGI(I,J)=0.0
      VSTAR(I,J)=VP(I,J)
    1 SCBETA(I,J)=0.0

      ITER=0
    1 CALL BKSCPE(NBLADE,M,VP,RC,SCBETA,F1,F2,FO,F1P,F2P,LMDM,TAU,DMY,
     1 A1,P,PP,CGIGMC,CETGMC)

      CALL VKASHI(D,NF,AI,CGIGMC,M,F1P,F2P,F1,F2,FO,EPS,THETA,RC,VP,
     1 NBLADE,VORTEX,SCBETA,DMY,CETGMC,TAU,A2,CLL,DTDGI,W,LMDM,P,PP,
     2 A1,A3)

      CALL SRDTBH(D,NF,AI,CGIGMC,M,F1P,F2P,F1,F2,FO,EPS,THETA,RC,VP,
     1 NBLADE,SCBETA,DMY,TAU,A2,CGI,CGIQC,SOURCE,VSTAR,CLL,DTDGI,W,LMDM,
     2 P,PP,A1,A3)

      CALL VSOURCE(D,NF,AI,CETQC,M,F1P,F2P,F1,F2,FO,EPS,THETA,RC,VP,
     1 NBLADE,LMDM,SCBETA,DMY,P,PP,TAU,A2,CGI,CGIQC,SOURCE,A1,A3)

      CALL BLADE(BTH,THETA,VSTAR,W,EPS,RC,CETQC,CETGMC,CLL,SCBETA,VP,
     1 CGIGMC,CGIQC,VORTEX,CGI,DTDGI,APSI,PSIMAX,SBETA,V,Y,NBLADE,DMY,
     2 ITER,SOURCE)
      IF(ITER.LT.10)GO TO 19
      STOP
      END
C*************************************************************************
C.....SUBROUTINE FOR CALCULATING LENGTHS OF STREAMLINES FROM LEADING ED
C*************************************************************************
      SUBROUTINE STRLM(X,Y,V,SBETA,NII)
      DIMENSION X(5,70),Y(5,70),V(5,70),SBETA(5,5),NII(5),AA(3),BB(3)
      Y(2,1)=42.5
      Y(3,1)=53.0
      Y(4,1)=60.5
C.....SECOND ORDER INTERPOLATION IS USED TO DECIDE CORDINATES OF LEADING
C.....EDGE FROM ONE KNOWN COORDINATE AT THE LEADING EDGE.
      DO 3 I=2,4
      YA=Y(I,1)
      DO 3 N=1,2
      DO 1 K=1,3
      JJ=15+K
      AA(K)=Y(I,JJ)
      BB(K)=X(I,JJ)
      IF(N.EQ.2)BB(K)=V(I,JJ)
    1 CONTINUE
      AO=BB(1)/((AA(1)-AA(2))*(AA(1)-AA(3)))
      A1=BB(2)/((AA(2)-AA(1))*(AA(2)-AA(3)))
      A2=BB(3)/((AA(3)-AA(1))*(AA(3)-AA(2)))
      BC=AO+A1+A2
      B1=-1.0*(AO*(AA(2)+AA(3))+A1*(AA(1)+AA(3))+A2*(AA(1)+AA(2)))
      B2=BB(1)-AO*AA(1)**2-B1*AA(1)
      IF(N.EQ.2)GO TO 2
      X(I,1)=AO*YA**2+B1*YA+B2
      GO TO 3
```

```
      [illegible]
      [illegible]
      [illegible]
```

```
      RTIO=(PI/2.)/ARTIP
      DO 22 I=1,5
      NB=III(I)
      DO 21 J=1,NB
      .
      .
      .
```

```
      RTIPS=RTIP*RTIP
      DO 21 I=1,3
      NB=NII(I)
      DO 21 J=1,NB
      SBETA(I,J)=SBETA(I,J)/RTIP
      Y(I,J)=Y(I,J)/RTIP
      X(I,J)=X(I,J)/RTIP
   21 Z(I,J)=Z(I,J)*RTIPS/6.2832
      RETURN
      END

C*****************************************************************
C     SUBROUTINE FOR CALCULATING LENGTH OF A CURVE BETWEEN ANY TWO POINTS,
C     A SECOND ORDER POLYNOMIAL PASSING THROUGH THREE POINTS IS ASSUMED
C     AS APPROXIMATING CURVE.
C*****************************************************************
      SUBROUTINE SSS(AA,BB)
      DIMENSION AA(3),BB(3)
      A0=AA(1)/((AA(1)-AA(2))*(AA(1)-AA(3)))
      A1=AA(2)/((AA(2)-AA(1))*(AA(2)-AA(3)))
      A2=AA(3)/((AA(3)-AA(1))*(AA(3)-AA(2)))
      A=A0+A1+A2
      B=B*(A*(AA(2)+AA(3))+A1*(AA(1)+AA(3))+A2*(AA(1)+AA(2)))
      IF(ABS(BB(3)).LE..1E-35)GO TO 1
      T1=2.*A*AA(1)+B1
      TS1=SQRT(1.+T1*T1)
```

```
      B=A(Z)+G1
      T(1.3+T2*T2)
      )*(T2*TS2+ALOG(T2+TS2)-T1*TS1-ALOG(T1+TS1))
      (K)
      T(1.3+G1*G1)*(AA(2)-AA(1))
      (K)
C*****************************************************************
C..... SUBROUTINE FOR CONVERTING THE INPUT VALUES OF SPECIFIED VELOCITY
C..... DISTRIBUTION TO NINETEEN POINTS AS DESIRED. SECOND ORDER INTERPOLA
C.....IS USED FOR THIS PURPOSE.
C*****************************************************************
      SUBROUTINE VGIVEN(CGI,GYL,XX,A1,NUMBER)
      DIMENSION XX(13),CGI(5,2,19),GYL(19),A1(19),AA(3),BB(3)
 2    FORMAT(24,11F10.5)
 3    FORMAT(24,*KNOWN VALUES ARE.....*)
 4    FORMAT(/ (6E20.8))
      WRITE(6,3)
      DO 11 JK=1,NUMBER
 11   WRITE(6,2)(XX(JK),((CGI(I,J,JK),J=1,2),I=1,5)
      DO 16 I=1,5
      DO 16 J=1,2
      DO 11 JB=1,NUMBER
 11   A1(JB)=CGI(I,J,JB)
      CGI(I,J,19)=A1(NUMBER)
      CGI(I,J,18)=A1(7)
      DO 15 JB=2,18
      FJ=GYL(JB)
      IF(JB.EQ.10)GO TO 15
      DO 14 JK=2,NUMBER
      IF(FJ.EQ.XX(JK))GO TO 13
      IF(FJ.LT.XX(JK-1))GO TO 14
      IF(FJ.GT.XX(JK))GO TO 14
      LAA=2
      IF(JK.EQ.NUMBER)LAA=3
      DO 12 L=1,3
      LA=JK+L-LAA
      AA(L)=XX(LA)
 12   BB(L)=A1(LA)
      E0=BB(1)/((AA(1)-AA(2))*(AA(1)-AA(3)))
      E1=BB(2)/((AA(2)-AA(1))*(AA(2)-AA(3)))
      E2=BB(3)/((AA(3)-AA(1))*(AA(3)-AA(2)))
      B0=E0+E1+E2
      B1=-(E0*(AA(2)+AA(3))+E1*(AA(1)+AA(3))+E2*(AA(1)+AA(2)))
      B2=BB(1)-B0*AA(1)**2-B1*AA(1)
      CGI(I,J,JB)=B0*FJ*FJ+B1*FJ+B2
      GO TO 15
 13   CGI(I,J,JB)=A1(JK)
      GO TO 15
 14   CONTINUE
 15   CONTINUE
```

```
      J=1,5
      WRITE(6,*)((CLL(I,J,K),K=1,19)
      STOP
      END
C*****************************************************************
C**** SUBROUTINE FOR INITIAL APPROXIMATION OF CAMBER LINE SHAPES.
C*****************************************************************
      SUBROUTINE SHAPE(SBETA,CGI,GYL,R,V,VORTEX,W,THETA,VP,RC,SCBETA,N,
     1 IMP,CLL)
      DIMENSION SBETA(5,50),CGI(5,2,19),GYL(19),R(5,70),V(5,70),
     1 W(5,19),N(5,19),VP(5,19),THETA(5,19),SCBETA(5,19),RC(5,19),
     2 EPS(5,19),CLL(5,19)
      WRITE(6,215,ERR=18.8)
      DO 2 J=1,19
      CLL(1,J)=1.75*GYL(J)
      CLL(2,J)=1.33*GYL(J)
      CLL(3,J)=1.05*GYL(J)
      CLL(4,J)=1.35*GYL(J)
      CLL(5,J)=0.65*GYL(J)
      DO 11 I=1,5
      DO 4 J=1,19
      VCGI(I,J)=CGI(I,2,J)-CGI(I,1,J)
      N(I,1)=0.0
      TANEPS=V(I,1)/V(I,1)
      TTN=TANEPS
      EPS(I,1)=ATAN(TANEPS)
      VP(I,1)=V(I,1)
      GAMA=0.0
      THETA(I,1)=0.0
      DSBETA=CLL(I,2)/SQRT(1.0+TANEPS**2)
      SCBETA(I,1)=0.0
      SCBETA(I,2)=DSBETA
      SSSSS=SCBETA(I,2)
      RC(I,1)=R(I,1)
      RC(I,2)=RADIUS(SSSSS,R,SBETA,I)
      DO 11 J=2,19
      J1=J-1
      GAMA=GAMA+FLOAT(M)*AREA(CLL,VORTEX,J1,J,I)
      N(I,J)=GAMA/(6.2832*RC(I,J))
      RAD=RC(I,J)
      VP(I,J)=VPPP(I,RAD,V,R)
      TANEPS=(N(I,M)+RC(I,J))/VP(I,J)
      EPS(I,J)=ATAN(TANEPS)
      IF(J.EQ.19)GO TO 10
      DSBETA=(CLL(I,J+1)-CLL(I,J))/SQRT(1.0+TANEPS**2)
      SCBETA(I,J+1)=SCBETA(I,J)+DSBETA
      SSSSS=SCBETA(I,J+1)
      RC(I,J+1)=RADIUS(SSSSS,R,SBETA,I)
C.....TRAPEZOIDAL RULE IS USED FOR INTEGRATION HERE FOR CONVINIENCE ONLY
   10 DT=(TTN/RC(I,J-1))+(TANEPS/RC(I,J))
      THETA(I,J)=THETA(I,J-1)+(DT/2.0)*(SCBETA(I,J)-SCBETA(I,J-1))
      TTN=TANEPS
   11 CONTINUE
      DO 12 I=1,5
```

```
      1     ...L,...J,CLL(I,J),GYL(J),SCBETA(I,J),THETA(I,J),RC(I,J),VP(I,J)
            ...
            ...
C*****...*****...*****...*****...*****...*****...*****...*****...*****...*****
C.....FUNCTION USED FOR SECOND ORDER INTERPOLATION AND/OR SELECTION OF
C.....VALUE OF R FOR ANY GIVEN VALUE OF SBETA.
C*****...*****...*****...*****...*****...*****...*****...*****...*****...*****
      FUNCTION RADIUS(S,R,SBETA,I)
      DIMENSION R(5,70),SBETA(5,50),AA(3),BB(3)
      DO 2 J=1,70
      IF(SBETA(I,J).GT.S)GO TO 2
      IF(SBETA(I,J+1).LT.S)GO TO 2
      IF(SBETA(I,J).EQ.S)GO TO 1
      DO 4 L=1,3
      LA=J+L-1
      AA(L)=SBETA(I,LA)
    4 BB(L)=R(I,LA)
      A0=BB(1)/((AA(1)-AA(2))*(AA(1)-AA(3)))
      A1=BB(2)/((AA(2)-AA(1))*(AA(2)-AA(3)))
      A2=BB(3) /((AA(3)-AA(1))*(AA(3)-AA(2)))
      B0=A0+A1+A2
      B1=-1.0*(A0*(AA(2)+AA(3))+A1*(AA(1)+AA(3))+A2*(AA(1)+AA(2)))
      B2=BB(1)-B0*AA(1)*AA(1)-B1*AA(1)
      RADIUS=B0*S*S+B1*S+B2
      GO TO 3
    1 RADIUS=R(I,J)
      GO TO 3
    2 CONTINUE
    3 RETURN
      END
C*****...*****...*****...*****...*****...*****...*****...*****...*****...*****
C.....FUNCTION USED FOR SECOND ORDER INTERPOLATION AND/OR SELECTION OF A
C.....VALUE OF VP FOR ANY GIVEN VALUE OF R.
C*****...*****...*****...*****...*****...*****...*****...*****...*****...*****
      FUNCTION VPPP(I,RAD,V,R)
      DIMENSION V(5,70),R(5,70),AA(3),BB(3)
      DO 2 J=1,70
      IF(R(I,J).GT.RAD)GO TO 2
      IF(R(I,J+1).LT.RAD)GO TO 2
      IF(R(I,J).EQ.RAD)GO TO 1
      DO 4 L=1,3
      LA=J+L-1
      AA(L)=R(I,LA)
    4 BB(L)=V(I,LA)
      A0=BB(1)/((AA(1)-AA(2))*(AA(1)-AA(3)))
      A1=BB(2)/((AA(2)-AA(1))*(AA(2)-AA(3)))
      A2=BB(3) /((AA(3)-AA(1))*(AA(3)-AA(2)))
      B0=A0+A1+A2
      B1=-1.0*(A0*(AA(2)+AA(3))+A1*(AA(1)+AA(3))+A2*(AA(1)+AA(2)))
      B2=BB(1)-B0*AA(1)*AA(1)-B1*AA(1)
      VPPP=B0*RAD*RAD+B1*RAD+B2
      GO TO 3
    1 VPPP=V(I,J)
      GO TO 3
```

```
C*********************************************************************
C.....SUBROUTINE USED FOR INTEGRATION BETWEEN ANY TWO POINTS HAVING INDEX
C.....NUMBER M. A SECOND ORDER POLYNOMIAL PASSING THROUGH THREE
C.....POINTS IS USED.
C*********************************************************************
      SUBROUTINE AREA(X,Y,A,M,I)
      DIMENSION X(5,19),Y(5,19)
      IF(M.EQ.1) GO TO 12
      H1=X(I,2)-X(I,1)
      H2=X(I,3)-X(I,2)
      H12=H1+H2
      A(I,1)=.16666667*(Y(I,1)*(H1*(2.0*H1+3.0*H2)/H12)+Y(I,2)*(H1*(3.0*
     1 H2+H1)/H2)-Y(I,3)*((H1**3)/(H2*H12)))
      GO TO 13
   12 H1=X(I,M)-X(I,M-1)
      H2=X(I,M)-X(I,M)
      H12=H1+H2
      A(I,M)=.16666667*(-Y(I,M-1)*((H2**3)/(H1*H12))+Y(I,M)*(H2*(3.0*H1+H
     1 2)/H1)+Y(I,M)*(H2*(3.0*H1+2.0*H2)/H12))
      RETURN
   13 RETURN
C*********************************************************************
C.....SUBROUTINE FOR CALCULATING VARIABLES USED IN THE WKB SOLUTION OF
C.....THE DIFFERENTIAL EQUATION.
C*********************************************************************
      SUBROUTINE WKBAPP(N,M,VP,RC,SCBETA,F1,F2,F0,F1P,F2P,LMDM,TAU,DMY,
     1 A1,P,PP,PPP,K)
      DIMENSION VP(5,19),RC(5,19),SCBETA(5,19),F1(5,M,19),F2(5,M,19),
     1 F0(5,M,19),F1P(5,M,19),F2P(5,M,19),P(5,19),LMDM(M),PP(5,19),
     2 PPP(5,19),K(5,19),KP(5,19),KPP(5,19),TAU(5,19),DMY(5,19),A1(19)
      REAL LMDM,K
C
    1 FORMAT(2X,*P(I,J) ARE.....*)
    2 FORMAT(2X,*LMDM(J) ARE.....*)
    3 FORMAT(2X,*K(I,J) ARE.....*)
    5 FORMAT(2X,6E20.8)
    6 FORMAT(2X,*DERIVATIVES OF P ARE.....*)
    8 FORMAT(2X,*F1(I,M,J) ARE.....*)
    9 FORMAT(2X,*DERIVATIVES OF F1(I,M,J) ARE.....*)
  101 FORMAT(2X,*F2(I,M,J) ARE.....*)
  102 FORMAT(2X,*DERIVATIVES OF F2(I,M,J) ARE.....*)
  103 FORMAT(2X,*F0(I,M,J) ARE.....*)
  104 FORMAT(2X,*TAU(I,J) ARE.....*)
  105 FORMAT(2X,*BETA(I,J) ARE.....*)
C
      PRINT 2
      DO 11 I=1,M
   11 LMDM(I)=I*N
      PRINT 5,(LMDM(I),I=1,M)
      DO 12 I=1,5
      DO 12 J=1,19
   12 P(I,J)=1./((RC(I,J)**2)*(VP(I,J)**2))
```

```
      DO [illegible] I=1,5
  [illegible]  PRINT 5,(P(I,J),J=1,19)
C.....VARIABLE DMY HERE IS BETA.
      DO [illegible] I=1,5
      DMY(I,1)=0.0
      DO [illegible] J=2,19
      J1=J-1
  [illegible]  DMY(I,J)=DMY(I,J1)+AREA(SCBETA,VP,J1,J,I)
      PRINT 104
      DO [illegible] I=1,5
  [illegible]  PRINT 5,(DMY(I,J),J=1,19)
      CALL DERIV(P,DMY,PP)
      CALL DERIV(PP,DMY,PPP)
      PRINT 6
      DO 103 I=1,5
  103 PRINT 5,(PP(I,J),J=1,19)
      PRINT 6
      DO 102 I=1,5
  102 PRINT 5,(PPP(I,J),J=1,19)
      DO 16 I=1,5
      DO 16 J=1,19
   16 K(I,J)=(4.0*P(I,J)*PPP(I,J)-5.0*PP(I,J)**2)/(16.0*P(I,J)**3)
      PRINT 3
      DO 104 I=1,5
  104 PRINT 5,(K(I,J),J=1,19)
C.....VARIABLE DMY HERE IS INTEGRAND FOR CALCULATING TAU.
      DO 13 I=1,5
      DO 13 J=1,19
   13 DMY(I,J)=1.0/RC(I,J)
      DO 14 I=1,5
      TAU(I,1)=0.0
      DO 14 J=2,19
      J1=J-1
   14 TAU(I,J)=TAU(I,J-1)+AREA(SCBETA,DMY,J1,J,I)
      PRINT 104
      DO 141 I=1,5
  141 PRINT 6,(TAU(I,J),J=1,19)
      DO 23 I=1,5
      A1(1)=0.0
      DO 17 NN=1,M
      B2=0.25*A1(1)/LMDM(NN)
      B3=0.25*K(I,1)/(LMDM(NN)**2)
C.....HERE ONLY EXPONTIAL IS STORED IN THE FOLLOWING VARIABLES.
      G1=B2-B3
      G2=-B2-B3
      TLM=LMDM(NN)*TAU(I,1)
      F1(I,NN,1)=G1+TLM
      F2(I,NN,1)=G2-TLM
   17 CONTINUE
C.....VARIABLE DMY HERE IS THE INTEGRAND FOR CALCULATING A1.
      DO 18 J=1,19
   18 DMY(I,J)=K(I,J)/RC(I,J)
      DO 19 J=2,19
      J1=J-1
```

```
      DO 1 J=1,19
      DO 1 I=1,5
    1 P(I,J)=P(I,J)/LMOM(PIA)
      DO 2 I=1,5
      DO 2 J=1,19
    2 T4U(I,J)=T4U(I,J)/(LMOM(PIA)**2)
      G1=-0.5
      G2=2.0-G1
      DO 3 I=1,5
      DO 3 J=1,19
      TLM=G1*T4U(I,J)
      F1P(I,J)=G1+TLM
    3 F2P(I,J)=G2-TLM
      DO 4 I=1,5
      DO 4 J=1,19
    4 PP(I,J)=-1.25*PP(I,J)*P(I,J)**(-1.25)
      DO 5 I=1,5
      DO 5 J=1,19
    5 P(I,J)=P(I,J)**(-0.25)
C.....CALCULATE DMY 4502 IS AGAIN BETA USED FOR DIFFERENTIATION.
      DO 24 I=1,5
      DMY(I,1)=0.0
      DO 24 J=2,19
      J1=J-1
   24 DMY(I,J)=DMY(I,J1)+AREA(SCBETA,VP,J1,J,I)
      CALL PRIMF(F1,DMY,F1P,M)
      CALL PRIMF(F2,DMY,F2P,M)
      DO 25 I=1,5
      DO 25 J=1,19
      DO 25 NN=1,M
   25 FO(I,NN,J)=F2P(I,NN,J)-F1P(I,NN,J)
      PRINT 8
      DO 26 I=1,5
      DO 26 J=1,19
   26 PRINT 5,(F1(I,NN,J),NN=1,M)
      PRINT 9
      DO 27 I=1,5
      DO 27 J=1,19
   27 PRINT 5,(F1P(I,NN,J),NN=1,M)
      PRINT 101
      DO 28 I=1,5
      DO 28 J=1,19
   28 PRINT 5,(F2(I,NN,J),NN=1,M)
      PRINT 102
      DO 29 I=1,5
      DO 29 J=1,19
   29 PRINT 5,(F2P(I,NN,J),NN=1,M)
      PRINT 103
      DO 30 I=1,5
      DO 30 J=1,19
   30 PRINT 5,(FO(I,NN,J),NN=1,M)
      RETURN
      END
C*************************************************************
C.... SUBROUTINE FOR CALCULATING DERIVATIVE OF A TWO DIMENSIONAL ARRAY.
C*************************************************************
      SUBROUTINE DERIV (Y,X,YP)
```

```
      DO 1 I=1,5
      H1=X(I,2)-X(I,1)
      H2=X(I,3)-X(I,2)
      H12=H1+H2
      B1=-(Y(I,1)/(H1*H12)-Y(I,2)/(H1*H2)    +Y(I,3)/(H2*H12)
      B2=-(Y(I,1)*(X(I,2)+X(I,3))/(H1*H12)-Y(I,2)*(X(I,1)+X(I,3))/(H1*H2
     1  )+Y(I,3)*(X(I,1)+X(I,2))/(H2*H12))
      YP(I,1)=(Y(I,2)-Y(I,1))/H1
      YP(I,2)=2.0*B0*X(I,2)  +B1
      DO 1 J=3,19
      H1=X(I,J)-X(I,J-1)
      H2=X(I,J+1)-X(I,J)
      H12=H1+H2
      B0=Y(I,J-1)/(H1*H12)-Y(I,J)/(H1*H2)+Y(I,J+1)/(H2*H12)
      B1=-(Y(I,J-1)*(X(I,J)+X(I,J+1))/(H1*H12)-Y(I,J)*(X(I,J-1)+X(I,J+1)
     1  )/(H1*H2)+Y(I,J+1)*(X(I,J-1)+X(I,J))/(H2*H12))
      YP(I,J)=2.0*B0*X(I,J)+B1
    1 CONTINUE
      YP(I,19)=(Y(I,19)-Y(I,18))/H2
   12 CONTINUE
      RETURN
      END
C*****************************************************************
C.....SUBROUTINE FOR CALCULATING DERIVATIVE OF A THREE DIMENSIONAL ARRAY
C*****************************************************************
      SUBROUTINE PRIME(Y,X,YP,M)
      DIMENSION Y(5,M,19),X(5,19),YP(5,M,19)
      DO 12 NN=1,M
      DO 12 I=1,5
      H1=X(I,2)-X(I,1)
      YP(I,NN,1)=(Y(I,NN,2)-Y(I,NN,1))/H1
      DO 11 J=2,16
      H1=X(I,J)-X(I,J-1)
      H2=X(I,J+1)-X(I,J)
      H12=H1+H2
      B0=Y(I,NN,J-1)/(H1*H12)-Y(I,NN,J)/(H1*H2)+Y(I,NN,J+1)/(H2*H12)
      B1=-(Y(I,NN,J-1)*(X(I,J)+X(I,J+1))/(H1*H12)-Y(I,NN,J)*(X(I,J-1)+X(
     1I,J+1)) /(H1*H2)+Y(I,NN,J+1)*(X(I,J-1)+X(I,J))/(H2*H12))
      YP(I,NN,J)=2.0*B0*X(I,J)+B1
   11 CONTINUE
      YP(I,NN,19)=(Y(I,NN,19)-Y(I,NN,18))/H2
   12 CONTINUE
      RETURN
      END
C*****************************************************************
C.....SUBROUTINE FOR CALCULATING INDUCED VELOCITIES DUE TO VORTICES
C.....PLACED ON THE CAMBER LINE.
C*****************************************************************
      SUBROUTINE YKASHID,NF,A1,CGIGMC,M,F1P,F2P,F1,F2,FD,EPS,THETA,RC,
     1 VP,N,VORTEX,SCBETA,DMY,CETGMC,TAU,BN,CLL,DTDGI,W,LMDM,P,PP,A1,A3)
      DIMENSION D(17,19),AI(19),CGIGMC(5,19),F1P(5,M,19),F2P(5,M,19),
     1 F1(5,M,19),F2(5,M,19),FD(5,M,19),EPS(5,19),LMDM(M),THETA(5,19),
     2 RC(5,19),VP(5,19),VORTEX(5,19),SCBETA(5,19),DMY(5,19),TAU(5,19),
     3 CETGMC(5,19),BN(19),W(5,19),DTDGI(5,19),CLL(5,19),P(5,19),
```

```
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      WRITE(6,*)' VALUES OF CGIGMC(I,J)  ARE.....................*)
      WRITE(6,*)' VALUES OF CETGMC(I,J)  ARE.....................*)

      [illegible]
      DO 11 I=1,17
      DO 11 J=1,19
   11 CGIGMC(I,J)=0.0
C.....CALCULATION OF SIN(N THETA)M  FOR FOURIER APPROXIMATION.
      DO 13 I=1,NF
      DO 13 J=2,16
      PHI=10.0*I*(J-1)
      IF(PHI.LT.360)GO TO 12
      IA=PHI/360.0
      PHI=PHI-IA*360.0
   12 PHI=PHI
   13 SIN(I,J)=SIN(PI*PHI/180.0)
   14 CONTINUE
      DO 15 I=1,5
      DO 15 J=1,19
      ONY(I,J)=0.0
   15 CGIGMC(I,J)=0.0
C.....VELOCITY INDUCED DUE TO VORTICES  IN GI   DIRECTION
      DO 24 I=1,5
      DO 24 J=1,19
      DO 22 JA=2,18
      ONY(I,JA)=0.0
      IF(JA.EQ.J)GO TO 22
      AJGMC=0.0
      DO 21 NN=1,M
      PHI=      LMDM(NN) *(THETA(I,J)-THETA(I,JA))
      FFF=(F1P(I,NN,JA)+PP(I,JA)/P(I,JA))*(F2P(I,NN,JA)+PP(I,JA)/P(I,JA)
     1)/F3(I,NN,JA)
      IF(JA.GE.J)GO TO 19
      FDYFP=EXP(F2(I,NN,J)-F2(I,NN,JA))*((P(I,J)*F2P(I,NN,J)+PP(I,J))/
     1 (P(I,JA)*F2P(I,NN,JA)+PP(I,JA)))
      FDYFP=EXP(F2(I,NN,J)-F2(I,NN,JA))*P(I,J)/(P(I,JA)*F2P(I,NN,JA)+
     1 PP(I,JA))
      GO TO 20
   19 FDYFP=EXP(F1(I,NN,J)-F1(I,NN,JA))*((P(I,J)*F1P(I,NN,J)+PP(I,J))/
     1 (P(I,JA)*F1P(I,NN,JA)+PP(I,JA)))
      FDYFP=EXP(F1(I,NN,J)-F1(I,NN,JA))*P(I,J)/(P(I,JA)*F1P(I,NN,JA)+
     1 PP(I,JA))
   20 T1=FDYFP*(COS(EPS(I,J))/LMDM(NN))*SIN(PHI)
      T2=FDYFP*(SIN(EPS(I,J))/(RC(I,J)*VP(I,J)))*COS(PHI)
      AJGMC=AJGMC+FFF*(T1+T2)
C.....CONVERGENCE CHECK OF INFINITE SERIES SUMMATION.
      CON=ABS(AJGMC*0.001)
      VAL=ABS(FFF*(T1+T2))
      IF(VAL.LE.CON)GO TO 211
   21 CONTINUE
```

```
C.....AN ANALOGY HERE IS THE INTEGRAND FOR CALCULATING CGIGMC.
      [illegible](I,JA)=VORTEX(I,JA)*AJGMC/(COS(EPS(I,JA)))
      DO 21 I=1,5
      AP=1.
      DO 22 JA=2,19
      J1=JA-1
   22 AP=AP+.5*(SCBETA,DMY,J1,JA,I)
      DMY(I,J)=AR*VP(I,J)
      DO 23 I=1,5
      AB(1)=2
      DO 23 I=1,5
   23 WRITE(1,(CAIGMC(I,J),J=1,19)
      DO 25 I=1,5
      DO 25 J=1,19
      DMY(I,J)=0.0
   25 CBETA(I,J)=0.0
C.....VELOCITY INDUCED DUE TO VORTICES  IN ETA  DIRECTION
      DO 33 I=1,5
      DO 261 JB=1,19
      AC=1.-2.*TAU(I,JB)/TAU(I,19)
C.....AC IS CAPITAL THETA APPEARING IN THE POISSON  INTEGRAL.
  261 AB(JB)=ARCOS(AC)
      DO 33 J=1,19
      DO 262 JA=1,19
  262 A1(JA)=0.0
      DO 30 JA=2,18
      IF(JA.EQ.J)GO TO 30
      AIGMC=0.0
      DO 29 NN=1,M
      PHI=       LMOM(NN) *(THETA(I,J)-THETA(I,JA))
      FFF=(F1P(I,NN,JA)+PP(I,JA)/P(I,JA))*(F2P(I,NN,JA)+PP(I,JA)/P(I,JA)
     1)/P0(I,NN,JA)
      IF(JA.GE.J)GO TO 27
      FPOYFP=EXP(F2(I,NN,J)-F2(I,NN,JA))*((P(I,J)*F2P(I,NN,J)+PP(I,J))/
     1  (P(I,JA)*F2P(I,NN,JA)+PP(I,JA)))
      FOYFP=EXP(F2(I,NN,J)-F2(I,NN,JA))*P(I,J)/(P(I,JA)*F2P(I,NN,JA)+
     1 PP(I,JA))
      GO TO 28
   27 FPOYFP=EXP(F1(I,NN,J)-F1(I,NN,JA))*((P(I,J)*F1P(I,NN,J)+PP(I,J))/
     1  (P(I,JA)*F1P(I,NN,JA)+PP(I,JA)))
      FOYFP=EXP(F1(I,NN,J)-F1(I,NN,JA))*P(I,J)/(P(I,JA)*F1P(I,NN,JA)+
     1  PP(I,JA))
   28 T1=FPOYFP*(SIN(EPS(I,J))/LMOM(NN))*SIN(PHI)
      T2=FOYFP*(COS(EPS(I,J))/(RC(I,J)*VP(I,J)))*COS(PHI)
      AIGMC=AIGMC+FFF*(T1-T2)
C.....CONVERGENCE CHECK OF INFINITE SERIES SUMMATION.
      CON=ABS(AIGMC*0.001)
      VAL=ABS(FFF*(T1-T2))
      IF(VAL.LE.CON)GO TO 291
   29 CONTINUE
  291 AIGMC=AIGMC*(NBLADE/PI)*MBLADE*(TAU(I,J)-TAU(I,JA))
C.....A1 IS THE UNEQUALLY SPACED FUNCTION TO BE REPRESENTED BY FOURIER
      A1(JA)=AIGMC*VORTEX(I,JA)*RC(I,JA)/COS(EPS(I,JA))
   30 CONTINUE                                    TO EQUALLY SPACED FUNCT
```

```
C.....[illegible] SECOND ORDER INTERPOLATION.
      DO [illegible]=2,18
      PJ=[illegible](J-1)*PI/18.0
      DO [illegible] JK=J,19
      IF(PJ.EQ.AA(JK))GO TO 301
      IF(PJ.LT.AA(JK-1))GO TO 302
      IF(PJ.GT.AA(JK))GO TO 302
      [illegible]
      IF(JK.EQ.19)LAA=3
      DO [illegible] L=1,3
      LA=JK+L-LAA
      AA(L)=AA(LA)
  [illegible] BB(L)=BB(LA)
      E=BB(1)/((AA(1)-AA(2))*(AA(1)-AA(3)))
      E1=BB(2)/((AA(2)-AA(1))*(AA(2)-AA(3)))
      E2=BB(3)/((AA(3)-AA(1))*(AA(3)-AA(2)))
      EC=E+E1+E2
      B1=-1.0*(E*(AA(2)+AA(3))+E1*(AA(1)+AA(3))+E2*(AA(1)+AA(2)))
      B2=BB(1)-B1*AA(1)**2-B1*AA(1)
      BMY(I,J)=EC*PJ*PJ+B1*PJ+B2
      GO TO 303
  301 BMY(I,J)=AI(JK)
      GO TO 303
  302 CONTINUE
  303 CONTINUE
      DO 31 NN=1,19
   31 BN(NN)=0.0
C.....CALCULATION OF FOURIER SERIES COEFFICIENTS.
      DO 33 NN=1,NF
      SUM=0.0
      DO 32 NI=2,18
   32 SUM=SUM+BMY(I,NI)*D(NN,NI)
      BN(NN)=(2.0/18.0)*SUM
   33 CONTINUE
C.....EVALUATION OF POISSON INTEGRAL USING THE PROCEDURE GIVEN IN CHAPTE
      AC=1.0-2.0*TAU(I,J)/TAU(I,19)
      NA=NF+1
      AI(1)=PI
      AI(2)=2.0*AC*AI(1)
      DO 36 JA=3,NA
      JA1=JA-1
      JA2=JA-2
   36 AI(JA)=2.0*AC*AI(JA1)-AI(JA2)
      SUM=0.0
      SUM=BN(2)*AI(1)-BN(NF-1)*AI(NF)-BN(NF)*AI(NA)
      NA=NF-1
      DO 37 JA=2,NA
      JA1=JA+1
      JA2=JA-1
   37 SUM=SUM+(BN(JA1)-BN(JA2))*AI(JA)
      SUM=0.5*SUM*(VP(I,J)/NBLADE)
      CETGMC(I,J)=SUM
   38 CONTINUE
      [illegible]
```

```
      IF(JA.EQ.1)GO TO 421
      DO 42 I=1,19
  421 AJQCYX=(A1(I,JA)*C/(PI*RC(I,JA)*VP(I,JA)))*NBLADE
      T=(TAU(I,J)-TAU(I,JA))/(2.*PI*RC(I,JA)*VP(I,JA)*(TAU(I,J)-TAU(I,JA)))
C.....DMY(I,J) IS THE INTEGRAND FOR EVALUATING CASCADE EFFECT.
      DMY(I,J)=SIN(EPS(I,JA)*(AJQCYX-T)
   42 CONTINUE
C.....INTEGRATION NEAR THE THE PRINCIPAL VALUE. UNEQUALLY SPACED DATA
C.....AROUND THE PRINCIPAL VALUE ARE FIRST CONVERTED TO EQUALLY SPACED
C.....DATA AND A LINEAR INTEGRATION IS USED NEAR PRINCIPAL VALUE,AND
C.....SECOND ORDER IN THE REST OF THE RANGE.
      AR=0.0
      DO 46 J=2,19
      J1=J-1
      JS=ABS(J-JA)
      IF(JS.GT.2)GO TO 459
      IF(J.EQ.1.OR.J.EQ.19)GO TO 458
      IF(J.EQ.JA)GO TO 46
      IF(J.EQ.(JA-1))GO TO 458
      CA=CLL(I,J+1)-CLL(I,J)
      CB=CLL(I,J)-CLL(I,J-1)
      IF(J.LT.JA)AR=AR+0.5*(DMY(I,J+1)*((CA**2)/CB+CA-CB)+DMY(I,J-1)*CB)
      IF(J.GT.JA)AR=AR+0.5*(DMY(I,J-1)*((CB**2)/CA+CB-CA)+DMY(I,J+1)*CA)
      IF(J.EQ.JA)AR=AR+0.5*(DMY(I,J-1)*CB+DMY(I,J+1)*CA)
      GO TO 46
  458 AR=AR+0.5*(CLL(I,JA)-CLL(I,J1))*(DMY(I,JA)+DMY(I,J1))
      GO TO 46
  459 AR=AR+AREA(CLL,DMY,J1,JA,I)
   46 CONTINUE
      CGI(CG(I,J)=AR*VP(I,J)
   47 CONTINUE
      DO 48 JB=1,19
   48 DMY(I,JB)=0.0
      DO 49 JA=2,18
      EPS=(SQRT(1.0+DTDGI(I,JA)**2)/2.0)*(CGI(I,1,JA)+CGI(I,2,JA))-
     1 VSTAR(I,JA)*COS(EPS(I,JA))-(W(I,JA)+RC(I,JA))*SIN(EPS(I,JA))-
     2 CGIOMC(I,JA)-CGICCG(I,JA)
      COSTH=1.0-2.0*TAU(I,JA)/TAU(I,19)
      SINTH=SQRT(1.0-COSTH**2)
C.....A1 IS THE FUNCTION APPEARING IN THE INTEGRAL EQUATION,TO BE REPRES
C.....BY FOURIER SERIES.
      A1(JA)=(EPS/(VP(I,JA)*COS(EPS(I,JA))))*SINTH
   49 CONTINUE
C.....CONVERSION OF UNEQUALLY SPACED FUNCTION A1 TO EQUALLY SPACED FUNCT
C.....DMY TO USE FOURIER APPROXIMATION .SECOND ORDER INTERPOLATION IS US
      DO 493 JB=2,18
      FJ=FLOAT(JB-1)*PI/18.0
      DO 492 JK=2,19
      IF(FJ.EQ.A3(JK))GO TO 491
      IF(FJ.LT.A3(JK-1))GO TO 492
      IF(FJ.GT.A3(JK))GO TO 492
      LAA=2
      IF(JK.EQ.19)LAA=3
      DO 4(JK=1,3
```

```
      S(I)=S(I-1)
   4  S(I)=AI(I-1)
      E0=AA(1)/((AA(1)-AA(2))*(AA(1)-AA(3)))
      E1=AA(2)/((AA(2)-AA(1))*(AA(2)-AA(3)))
      E2=AA(3)/((AA(3)-AA(1))*(AA(3)-AA(2)))
      E=E0+E1+E2
      B1=-1.0*(E0*(AA(2)+AA(3))+E1*(AA(1)+AA(3))+E2*(AA(1)+AA(2)))
      B2=E0*AA(1)**2-B1*AA(1)
      AY(I,J)=C0*FJ*FJ+C1*FJ+C2
      GO TO 43
   41 AY(I,J)=A2(JK)
      GO TO 43
   42 GO TO 43
   43 CONTINUE
      DO 9 I=1,19
    9 A(I,J)=0.0
C.....CALCULATION OF FOURIER COEFFICIENTS
      DO 52 NN=1,NF
      SUM=0.0
      DO 51 NI=2,18
   51 SUM=SUM+AMY(I,NI)* D(NN,NI)
   52 BN(NN)=(2.0/18.0)*SUM
      ERROR=0.0
C.....EVALUATION OF POISSON INTEGRAL.
      DO 57 J=2,18
      AC=1.0-2.0*TAU(I,J)/TAU(I,19)
      NA=NF+1
      AI(1)=PI
      AI(2)=2.0*AC*AI(1)
      DO 55 JA=3,NA
      JA1=JA-1
      JA2=JA-2
   55 AI(JA)=2.0*AC*AI(JA1)-AI(JA2)
      SUM=0.0
      SUM=BN(2)*AI(1)-BN(NF-1)*AI(NF)-BN(NF)*AI(NA)
      NA=NF-1
      DO 56 JA=2,NA
      JA1=JA+1
      JA2=JA-1
   56 SUM=SUM+(BN(JA1)-BN(JA2))*AI(JA)
      COSTH=1.0-2.0*TAU(I,J)/TAU(I,19)
      SINTH=SQRT(1.0-COSTH**2)
      SUM=-(SUM*VP(I,J)*CCS(EPS(I,J)))/(PI*SINTH)
      SAVSRS=SOURCE(I,J)
      ER=ABS((SAVSRS-SUM)/SUM)*100.0
C.....OVER-RELAXATION OF THE SOLUTION OF INTEGRAL EQUATION USING
C.....VARIABLE DAMPING FACTOR.
      IF(ER.LT. 5.)SOURCE(I,J)=SAVSRS+1.75*(SUM-SAVSRS)
      IF(ER.GE. 5..AND.ER.LT.10.)SOURCE(I,J)=SAVSRS+1.70*(SUM-SAVSRS)
      IF(ER.GE.10..AND.ER.LT.20.)SOURCE(I,J)=SAVSRS+1.65*(SUM-SAVSRS)
      IF(ER.GE.20..AND.ER.LT.30.)SOURCE(I,J)=SAVSRS+1.60*(SUM-SAVSRS)
      IF(ER.GE.30.)SOURCE(I,J)=SAVSRS+1.55*(SUM-SAVSRS)
      IF(ITER.LE.2)SOURCE(I,J)=SUM
      IF(ABS(SOURCE(I,J)).LT..1E-10)GO TO 57
```

```
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
C..... [illegible]
C..... [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
      [illegible]
C**********************************************************************
C..... SUBROUTINE FOR CALCULATION OF INDUCED VELOCITIES DUE TO SOURCE/SIN
C..... INTRIBUTION PLACED ON THE CAMBER LINE AT EACH STREAM SURFACE.
C**********************************************************************
      SUBROUTINE VSOURC(D,NF,AI,CETQC ,M,F1P,F2P,F1,F2,FD,EPS,THETA,RC,
     1 VP,N,LMDM,SCBETA,DMY,P,PP,TAU,BN,CGI,CGIQC,SOURCE,A1,A3)
      DIMENSION D(17,19),AI(19),CETQC(5,19),F1P(5,M,19),F2P(5,M,19),
     1 F1(5,M,19),F2(5,M,19),FD(5,M,19),LMDM(M),EPS(5,19),THETA(5,19),
     2 RC(5,19),VP(5,19),    PP(5,19),SCBETA(5,19),DMY(5,19),TAU(5,19),
     3 SOURCE(5,19),BN(19),CGI(5,2,19),CGIQC(5,19),P(5,19)
     4,A1(19),A3(19),AA(3),BB(3)
C
      REAL LMDM,NBLADE
      NBLADE=4
      PI=ARSIN(1.0)*2.0
    1 FORMAT(2X,6E20.8)
    8 FORMAT(2X,*VALUES OF CGIQC(I,J) ARE.....*)
    9 FORMAT(2X,*VALUES OF CETQC(I,J) ARE.....*)
C
C..... VELOCITY INDUCED DUE TO SOURCES IN  GI  DIRECTION
      DO 60 I=1,5
      DO 60 J=1,19
      DMY(I,J)=0.0
   60 CGIQC(I,J)=0.0
      DO 72 I=1,5
      DO 601 JB=1,19
      AC=1.0-2.0*TAU(I,JB)/TAU(I,19)
C..... A3 IS CAPITAL THETA APPEARING IN POISSON INTEGRAL
  601 A3(JB)=ARCOS(AC)
      DO 72 J=1,19
      DO 602 JB=1,19
  602 A1(JB)=0.0
      DO 64 JA=2,18
      IF(JA.EQ.J)GO TO 64
      AJOC=0.0
      DO 63 NN=1,M
      PH1=LMDM(NN)*(THETA(I,J)-THETA(I,JA))
      IF(ABS(PH1).GT.1.0)GO TO 61
```

```
      1 P(I,JA)
        F2=(F2(I,NN,J)-F2(I,NN,JA))*(P(I,J)/P(I,JA))
        F1=(F1(I,NN,J)-F1(I,NN,JA))*(P(I,J)*F1P(I,NN,J)+PP(I,J))/
      1 P(I,JA)
        F1=(F1(I,NN,J)-F1(I,NN,JA))*(P(I,J)/P(I,JA))
        T1=(-DMY*COS(EPS(I,J))*COS(PHI)
        T2=(-DMY/(RC(I,J)*VP(I,J)))*LMOM(NN)*SIN(EPS(I,J))*SIN(PHI)
        AJAC=AJAC+(T1-T2)/FC(I,NN,JA)
C.....RELATIVE CONVERGENCE CHECK OF INFINITE SERIES SUMMATION.
        CC=ABS(AJAC*.001)
        AAL=ABS((T1-T2)/FC(I,NN,JA))
        IF(AAL.LE.CON)GO TO 631
  630   CONTINUE
  631   AJAC=AJAC*(NBLADE/(PI*RC(I,JA)*VP(I,JA)))*NBLADE*(TAU(I,J)
      1-TAU(I,JA))
C.....A1 IS THE FUNCTION TO BE REPRESENTED BY FOURIER SERIES.
        A1(JA)=AJAC*SOURCE(I,JA)*RC(I,JA)/COS(EPS(I,JA))
  640   CONTINUE
C.....CONVERSION OF UNEQUALLY SPACED DATA A1 TO EQUALLY SPACED DATA DMY
C.....USING SECOND ORDER INTERPOLATION.
        DO 643 JB=2,18
        FJ=FLOAT(JB-1)*PI/18.0
        DO 642 JK=2,19
        IF(FJ.EQ.A3(JK))GO TO 641
        IF(FJ.LT.A3(JK-1))GO TO 642
        IF(FJ.GT.A3(JK))GO TO 642
        LAA=2
        IF(JK.EQ.19)LAA=3
        DO 644 L=1,3
        LA=JK+L-LAA
        AA(L)=A3(LA)
  644   BB(L)=A1(LA)
        E0=BB(1)/((AA(1)-AA(2))*(AA(1)-AA(3)))
        E1=BB(2)/((AA(2)-AA(1))*(AA(2)-AA(3)))
        E2=BB(3)/((AA(3)-AA(1))*(AA(3)-AA(2)))
        B0=E0+E1+E2
        B1=-1.0*(E0*(AA(2)+AA(3))+E1*(AA(1)+AA(3))+E2*(AA(1)+AA(2)))
        B2=BB(1)-B0*AA(1)**2-B1*AA(1)
        DMY(I,JB)=B0*FJ*FJ+B1*FJ+B2
        GO TO 643
  641   DMY(I,JB)=A1(JK)
        GO TO 643
  642   CONTINUE
  643   CONTINUE
        DO 65 NN=1,19
   65   BN(I,NN)=0.0
C.....EVALUATION OF FOURIER COEFFICIENTS.
        DO 67 NN=1,NF
        SUM=0.0
        DO 66 NI=2,18
   66   SUM=SUM+DMY(I,NI)* D(NN,NI)
        BN(I,NN)=(2.0/18.0)*SUM
```

```
      C=1.-2.*(TAU(I,J)-TAU(I,1))/(TAU(I,19)-TAU(I,1))
      NA=NA+1
      AI(1)=0.
      AI(NA)=2.*ACOS(C)
      DO 71 JA=2,NA
      JA1=JA-1
      JA2=JA-2
   7  AI(JA)=2.*ACOS(C)+AI(JA1)-AI(JA2)
      SUM=0.
      BSUM=BN(2)*AI(1)-BN(NF-1)*AI(NF)-BN(NF)*AI(NA)
      NA=NA-1
      DO 71 JA=2,NA
      JA1=JA+1
      JA2=JA-1
  71  SUM=BSUM+(BN(JA1)-BN(JA2))*AI(JA)
      SUM=0.5*SUM*(VP(I,J)/NBLADE)
      CGI(C(I,J)=SUM
  72  CONTINUE
      WRITE(3,3)
      DO 73 I=1,5
  73  WRITE(3,4)(CGI(C(I,J),J=1,19)
C..... VELOCITY INDUCED DUE TO SOURCES IN ETA  DIRECTION
      DO 731 I=1,5
      DO 731 J=1,19
      DNY(I,J)=0.0
 731  CETOC(I,J)=0.0
      DO 77 I=1,5
      DO 77 J=1,19
      DO 77 JA=2,18
      DNY(I,JA)=0.0
      IF(JA.EQ.J) GO TO 77
      AIOC=0.0
      DO 76 NN=1,M
      PHI=LMDM(NN)*(THETA(I,J)-THETA(I,JA))
      IF(JA.GE.J)GO TO 74
      FPOYF =EXP(F2(I,NN,J)-F2(I,NN,JA))*(P(I,J)*F2P(I,NN,J)+PP(I,J))/
     1 P(I,JA)
      FOYF=EXP(F2(I,NN,J)-F2(I,NN,JA))*(P(I,J)/P(I,JA))
      GO TO 75
  74  FPOYF =EXP(F1(I,NN,J)-F1(I,NN,JA))*(P(I,J)*F1P(I,NN,J)+PP(I,J))/
     1 P(I,JA)
      FOYF=EXP(F1(I,NN,J)-F1(I,NN,JA))*(P(I,J)/P(I,JA))
  75  T1=FPOYF*SIN(EPS(I,J))*COS(PHI)
      T2=(FOYF/(RC(I,J)*VP(I,J)))*LMDM(NN)*COS(EPS(I,J))*SIN(PHI)
      AIOC=AIOC+(T1+T2)/FC(I,NN,JA)
C.....CONVERGENCE CHECK OF INFINITE SERIES SUMMATION.
      CON=ABS(AIOC*0.001)
      VAL=ABS((T1+T2)/FC(I,NN,JA))
      IF(VAL.LE.CON)GO TO 761
  76  CONTINUE
 761  AIOC=-(AIOC*NBLADE)/(PI*RC(I,JA)*VP(I,JA))
      DNY(I,JA)=AIOC*SOURCE(I,JA)/COS(EPS(I,JA))
  77  CONTINUE
      WRITE(6,*)' EVALUATION OF INTEGRAL'
```

```
      DO 7 J=2,19
      J1=J-1
    7 SCBETA(I,J)=SCBETA(I,J1)+DMY(I,J)
      VP(I,J)=W*V(I,J)
    6 CONTINUE
      ITER=0
      DO 8 I=1,5
    8 WRITE(6,9)I,(SCBETA(I,J),J=1,19)
      WRITE(6,10)
C*****************************************************************
C.....SUBROUTINE FOR CALCULATION OF CAMBER LINE SHAPE,BLADE THICKNESS AN
C.....MODIFICATION OF VSTAR TO INCLUDE THE EFFECT OF THICKNESS.
C*****************************************************************
      SUBROUTINE BLADE(BTH,THETA,VSTAR,W,EPS,RC,CETQC,CETGMC,CLL,SCBETA,
     1 DTDGI,CGIGMC,CGIQC,VORTEX,CGI,DTDGI,APS,PSIMAX,SBETA,V,R,N,DMY,ITER,
     2 SAVBTH,STHETA)
      DIMENSION BTH(5,19),SAVBTH(5,19),STHETA(5,19),THETA(5,19),
     1 VSTAR(5,19),W(5,19),EPS(5,19),RC(5,19),CETQC(5,19),CETGMC(5,19),
     2 CLL(5,19),SCBETA(5,19),VP(5,19),CGIGMC(5,19),CGIQC(5,19),APS(5),
     3 VORTEX(5,19),CGI(5,2,19),DTDGI(5,19),DRW(5),AK(5),ALPHA(5),
     4 PS(5),PKH(5),SBETA(5,50),R(5,70),V(5,70),DMY(5,19),RW(5),
     5 SPACE(5,19)
C
    1 FORMAT(8E20.8)
    2 FORMAT(2X,*ITERATION NO.*,I5,*,.....*)
    3 FORMAT(2X,*THETA(I,J)  AND  SCBETA(I,J)  FOR CAMBER LINE ARE....*)
    4 FORMAT(2X,*BLADE PROFILE THICKNESS BTH(I,J) .....*)
    5 FORMAT(2X,*VSTAR(I,J) AND VP(I,J) ARE .....*)
    6 FORMAT(2X,*ITERATION NO.*,I5,*,CUMULATIVE PERCENTAGE CHANGE IN THE
     1TA(I,J)=*,E16.8)
    7 FORMAT(2X,*ITERATION NO.*,I5,*,CUMULATIVE PERCENTAGE CHANGE IN BLA
     1DE PROFILE THICKNESS IS=*,E16.8)
    8 FORMAT(2X,*BLADE PROFILE CONVERGED IN*,I5,*,ITERATIONS*)
    9 FORMAT(2X,*EPS(I,J)  ARE.....*)
C
      N=PSIMAX/(4.0*6.2832)
      ITER=ITER+1
      PRINT 2,ITER
      DO 11 I=1,5
      DO 11 J=1,19
      DMY(I,J)=0.0
      SAVBTH(I,J)=BTH(I,J)
   11 STHETA(I,J)=THETA(I,J)
C.....CAMBER LINE SHAPE
      DO 16 I=1,5
      W(I,1)=0.0
C.....SOLUTION OF QUADRATIC EQUATION IN SIN(EPS) FOR CALCULATION OF EPS.
      F=VSTAR(I,1)/RC(I,1)
      G=-(CETQC(I,1)+CETGMC(I,1))/RC(I,1)
      FF=1.0+F**2
      B=(2.0*F*G)/FF
      C=(G**2-1.0)/FF
      D=B**2-4.0*C
```

```
      R1=(-B+SQRT(CHECK))
      R2=(-B-SQRT(CHECK))
      IF((ABS(R1).GT.1.0).OR.(R1.LT.0.0))GO TO 36
      IF((ABS(R2).GT.1.0).OR.(R2.LT.0.0))GO TO 37
      D1=ARSIN(R1)-EPS(I,2)
      D2=ARSIN(R2)-EPS(I,2)
      IF(ABS(D1).LT.ABS(D2))GO TO 38
      EPS(I,1)=ARSIN(R2)
      GO TO 39
   38 EPS(I,1)=ARSIN(R1)
      GO TO 39
   36 IF((ABS(R2).GT.1.0).OR.(R2.LT.0.0))GO TO 35
      EPS(I,1)=ARSIN(R2)
      GO TO 39
   37 EPS(I,1)=ARSIN(R1)
   39 ANY(I,1)=TAN(EPS(I,1))/RC(I,1)
      GAMA=0.0
      OSBETA=CCL(I,2)/SQRT(1.0+TAN(EPS(I,1))**2)
      SBETA(I,2)=OSBETA
      SSSSS=SBETA(I,2)
      RC(I,2)=CAMBUS(SSSSS,R,SBETA,I)
      DO 14 J=2,19
      J1=J-1
      GAMA=GAMA+FLOAT(2)*AREA(CLL,VORTEX,J1,J,I)
      W(I,J)=GAMA/(6.2832*RC(I,J))
      RAD=RC(I,J)
      VP(I,J)=VPPP(I,RAD,V,R)
      IF(ITER.EQ.1)VSTAR(I,J)=VP(I,J)
      B=VSTAR(I,J)/(W(I,J)+RC(I,J))
      C=-(CETGC(I,J)+CETGMC(I,J))/(W(I,J)+RC(I,J))
      FF=1.0+B**2
   31 PRINT1,(SBETA(I,J),J=1,19)
C.....SLAVE  PROFILE  THICKNESS
      DO 17 I=1,2
      B=(2.0*B*C)/FF
      C=(C**2-1.0)/FF
      CHECK=B**2-4.0*C
      IF(CHECK.LT.0.0)GO TO 39
      R1=0.5*(-B+SQRT(CHECK))
      R2=0.5*(-B-SQRT(CHECK))
      IF((ABS(R1).GT.1.0).OR.(R1.LT.0.0))GO TO 40
      IF((ABS(R2).GT.1.0).OR.(R2.LT.0.0))GO TO 41
      D1=ARSIN(R1)-EPS(I,J-1)
      D2=ARSIN(R2)-EPS(I,J-1)
      IF(ABS(D1).LT.ABS(D2))GO TO 42
      EPS(I,J)=ARSIN(R2)
      GO TO 43
   42 EPS(I,J)=ARSIN(R1)
      GO TO 43
   40 IF((ABS(R2).GT.1.0).OR.(R2.LT.0.0))GO TO 39
      EPS(I,J)=ARSIN(R2)
      GO TO 43
   41 EPS(I,J)=ARSIN(R1)
      GO TO 43
```

```
                                    TO 15
                  (EPS(I,J)+EPS(I,J+1))
                  (CLL(I,J+1)-CLL(I,J))/SQRT(1.0+TAN(TTH)**2)
              (I,J+1)=SCRETA(I,J)+DSBETA
              (I,J+1)
              (SSSSS,DMY,SCRETA,I)
              (EPS(I,J+1))/RC(I,J+1)
              -TAN(EPS(I,J))/RC(I,J)
              (I,J-1)+AREA(SCRETA,DMY,J1,J,I)
              3
              7
              I-1,5
              I,(EPS(I,J),J=1,19)
              3
              I=1,5
              1,(THETA(I,J),J=1,19)
              J=1,19
              (I,J)=SOURCE(I,J)/(RC(I,J)*VP(I,J))
              I=1,5
              =1,5
              J=2,19
              J1=J-1
              =AR+AREA(CLL,DMY,J1,J,I)
              =VSTAR(I,J)*COS(EPS(I,J))*(W(I,J)+RC(I,J))*SIN(EPS(I,J))+
            CGIOC(I,J)+CGIOC(I,J)
              (I,J)=1.5*AR*RC(I,J)*VP(I,J)/ACGIC
              CONTINUE
              4
              I=1,5
              I,(BTH(I,J),J=1,19)
C.....CALCULATION OF DTDGI.
              DERIV(BTH,CLL,DTDGI)
              I=1,5
              J=1,19
              VORTEX(I,J)=SQRT(1.0+DTDGI(I,J)**2)*(CGI(I,2,J)-CGI(I,1,J))
C.....
C.....CHECK FOR CONVERGENCE.
C.....
C.....CAMBER LINE SHAPE CONVERGENCE.
              ERROR=0.0
              I=1,5
              J=1,19
              IF(ABS(THETA(I,J)).LT..1E-05)GO TO 28
              ERROR=ERROR+ABS((THETA(I,J)-STHETA(I,J))*100.0/THETA(I,J))
              CONTINUE
              ITER,ERROR
              IF(ERROR-10.0)29,27,27
C.....BLADE PROFILE THICKNESS CONVERGENCE.
              ERROR=0.0
              I=1,5
              J=1,19
              IF(ABS(BTH(I,J)).LT..1E-05)GO TO 30
              ERROR=ERROR+ABS((BTH(I,J)-SAVBTH(I,J))*(100.0/BTH(I,J)))
              CONTINUE
```

```
      DO [illegible] I=1,27,27
      [illegible]
C.....[illegible] OF VALUES ON EACH STREAM SURFACES.
      [illegible]
      DO 1 I=2,
      [illegible]S(I)/6.2832
      [illegible](I)=0.
      AK(1)=0.
      AK(I)=0.
      DW(I)=0.
      DO 2 I=1,19
      DO 2 J=1,5
    2 RW(I)=RW(I)+DRW(I,J)*W(I,J)
      DW(1)=(RW(2)-RW(1))/H
      DW(2)=(RW(3)-RW(1))/(2.0*H)
      DW(3)=(RW(4)-RW(2))/(2.0*H)
      DW(4)=(RW(5)-RW(3))/(2.0*H)
      DW(5)=(RW(5)-RW(4))/H
      DO 21 I=1,5
   21 FK(I)=(-1.-W(I,J)/RC(I,J))*DRW(I)/(VP(I,J)**2)
C.....INTEGRATION FORMULAS FOR EQUALLY SPACED DATA USED BELOW ARE OF SEC
C.....H IS THE SPACING OF INDEPENDENT VALIABLE(ALPHA).
      AK(1)=0.0
      AK(2)=(H/12.0)*(5.0*FK(1)+8.0*FK(2)-FK(3))
      DO 22 I=3,5
   22 AK(I)=AK(I-1)+(H/12.0)*(5.0*FK(I)+8.0*FK(I-1)-FK(I-2))
      DO 23 I=1,5
   23 FKH(I)=AK(I)
      AKH=(H/3.0)*(FKH(1)+4.0*FKH(2)+2.0*FKH(3)+4.0*FKH(4)+FKH(5))
      AKH=-(6.2832/PSIMAX)*AKH
      AK(1)=AKH
      DO 24 I=2,5
   24 AK(I)=AK(I)+AKH
      DO 25 I=1,5
   25 VSTAR(I,J)=VP(I,J)*(1.0+AK(1))
   26 CONTINUE
      PRINT 5
      DO 33 I=1,5
      PRINT 1,(VSTAR(I,J),J=1,19)
   33 PRINT 1,(VP(I,J),J=1,19)
      RETURN
      END

      0000
```

ENCOUNTERED ON SYSTEM INPUT FILE.

TICS =

```
      000007
      000000
      001309
      001597
      000000
```

TED
HED

A 54001

Date Slip A 54001

This book is to be returned on the date last stamped.

CD 6 72.9

ME-1977-M-JAI-NUM